RECONSTRUCTION OF 3D NEURONAL STRUCTURES FROM DENSELY PACKED ELECTRON MICROSCOPY DATA STACKS

A Dissertation

by

HUEI-FANG YANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2011

Major Subject: Computer Science

RECONSTRUCTION OF 3D NEURONAL STRUCTURES FROM DENSELY PACKED ELECTRON MICROSCOPY DATA STACKS

A Dissertation

by

HUEI-FANG YANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yoonsuck Choe
Committee Members,	John Keyser
	Frank Shipman
	Louise C. Abbott
Head of Department,	Duncan M. (Hank) Walker

August 2011

Major Subject: Computer Science

ABSTRACT

Reconstruction of 3D Neuronal Structures from Densely Packed Electron Microscopy Data Stacks. (August 2011)

Huei-Fang Yang, B.Ed., National Taiwan Normal University, Taiwan;

M.Ed., National Taiwan Normal University, Taiwan

Chair of Advisory Committee: Dr. Yoonsuck Choe

The goal of fully decoding how the brain works requires a detailed wiring diagram of the brain network that reveals the complete connectivity matrix. Recent advances in high-throughput 3D electron microscopy (EM) image acquisition techniques have made it possible to obtain high-resolution 3D imaging data that allows researchers to follow axons and dendrites and to identify pre-synaptic and post-synaptic sites, enabling the reconstruction of detailed neural circuits of the nervous system at the level of synapses. However, these massive data sets pose unique challenges to structural reconstruction because the inevitable staining noise, incomplete boundaries, and inhomogeneous staining intensities increase difficulty of 3D reconstruction and visualization.

In this dissertation, a new set of algorithms are provided for reconstruction of neuronal morphology from stacks of serial EM images. These algorithms include (1) segmentation algorithms for obtaining the full geometry of neural circuits, (2) interactive segmentation tools for manual correction of erroneous segmentations, and (3) a validation method for obtaining a topologically correct segmentation when a set of segmentation alternatives are available. Experimental results obtained by using EM images containing densely packed cells demonstrate that (1) the proposed segmentation methods can successfully reconstruct full anatomical structures from EM

images, (2) the editing tools provide a way for the user to easily and quickly refine incorrect segmentations, (3) and the validation method is effective in combining multiple segmentation results. The algorithms presented in this dissertation are expected to contribute to the reconstruction of the connectome and to open new directions in the development of reconstruction methods.

To my family

ACKNOWLEDGMENTS

My most sincere thanks go to my committee chair, Dr. Yoonsuck Choe, who has guided me through every stage of my Ph.D. study. I am especially thankful for his advice, encouragement, and patience which has helped me accomplish my dreams. I would also like to express my gratitude to my committee members, Dr. John Keyser, Dr. Frank Shipman, and Dr. Louise Abbott, for their helpful advice and suggestions, which were precious to me.

Deep appreciation also goes to my best friends, Dr. Wen-Chun Chen, Ms. Yen-Lin Chen, Ms. Mu-Fen Hsieh, and Ms. Hsiu-Ping Li. Their help, generosity, and friendship have made my life at Texas A&M University pleasant. I would also like to warmly acknowledge my lab mates, Mr. Ji Ryang Chung and Mr. Timothy Mann, for their valuable discussions during my pursuit of knowledge.

Finally, I would like to dedicate this dissertation to my dearest mother, sisters, and brother for their unconditional support and everlasting love in every possible way. I could not have completed it without them.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Resolving fundamental questions of how the brain works is undoubtedly one of the greatest challenges in neuroscience research. To overcome this challenge, neuroscientists are dedicated to the study of the brain function for decades. However, the goal of fully decoding how the brain works will not be accomplished without a detailed wiring diagram of the brain network that reveals the complete connectivity matrix. A wiring diagram is necessary and crucial for interpreting and understanding neural computations [1]. Recently, because of the availability of high-resolution imaging data, reconstruction of neural circuits of the brain has been receiving significant attention, and many methods for such a task are currently under development. This dissertation aims to provide computational solutions to the task of reconstructing detailed neural connectivity from large-scale and high-resolution data sets.

A. Motivation: Connectomics

The human brain is a complex biological structure, which consists of about 100 billion neurons and 100 trillion synaptic connections and thus forms a highly complex and intertwined network [2]. It is widely believed that the functioning of the brain is highly correlated to its anatomical structures and the dynamics of the network. To better understand how the interconnections of neurons account for the function and dysfunction of the brain, researchers have put in a great amount of effort to map out the full neural circuits of the nervous system. Indeed, if a complete and detailed wiring diagram of the nervous system at the level of synapses, i.e. the connectome [3–

This dissertation follows the style of *IEEE Transactions on Medical Imaging*.

5], is reconstructed, it will provide valuable insight to understand how the electrical signals flow within the brain and how the activities of the brain govern behaviors and will also greatly help explain the course of development of brain disorders and diseases [6–9].

Accomplishing the goal of obtaining a physical map of neural circuits requires data acquisition techniques that are capable of imaging tissues at sufficiently high resolution. In the past few years, advances in high-throughput three-dimensional (3D) image acquisition techniques [1, 10–17] have made it possible to obtain high-resolution imaging data, which brings promising prospects for the reconstruction of the connectome.

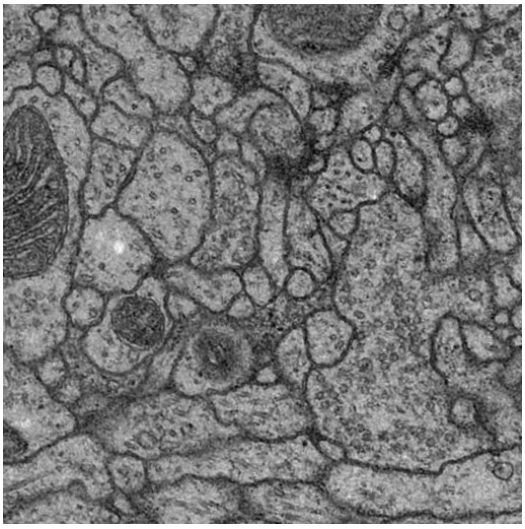
The oldest imaging technique that provides sufficient resolution to follow each neurite can be traced back to light microscopy (LM), which offers a reconstruction of sparsely labeled neurites [12]. This approach may not reveal each neuron's characterization and functionality as well as its connectivity with others. Knife-Edge Scanning Microscopy (KESM) [18, 14, 19] is a high-throughput data acquisition technique that images large volumes of tissues (1 cm^3) at the sub-micrometer scale. In KESM, the x-y resolution of an image is $0.3-0.6$ μm/pixel with a z resolution of $0.5-1$ μm/pixel. The KESM imaging data is important for understanding the complex network of the entire brain at sub-cellular resolution.

However, to fulfil the goal of reconstructing the connectomes that reveals synaptic connections, the tissue has to be imaged at a much higher resolution. Given such high imaging resolution, electron microscopy (EM) has been offering 3D imaging data that enables the reconstruction of neural circuits for the study of neuroanatomy at nanometerscale for years [20, 21]. A prominent example is that of the entire nervous system of the nematode *Caenorhabditis elegans*, a total of 302 neurons and over $7,000$ synaptic connections, which was completely reconstructed from serial section Trans-

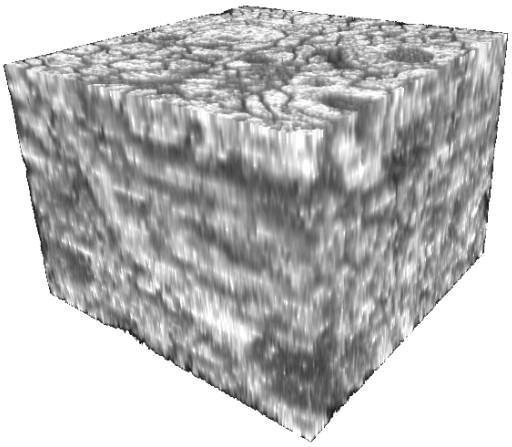
mission Electron Microscopy (ssTEM) imaging data [22]. This is the only connectome available to date since the seminal work of White et al. back in the mid 1980's. Over the past several years, development of various electron microscopy (EM) techniques, such as ssTEM [11, 20, 21], Serial Block Face Scanning Electron Microscopy (SBF-SEM) [10], and Automatic Tape-collecting Lathe Ultramicrotome (ATLUM) [23], has opened new roads for the reconstruction of densely labeled neurites. The EM imaging approaches provide 3D data at a resolution that allows researchers to follow axons and dendrites and to identify pre-synaptic and post-synaptic connectivity [1, 12]. For instance, in SBFSEM, the lateral (x-y) resolution can be as small as $10-20$ nm/pixel, and the sectioning thickness (z resolution) is around 30 nm.

B. The Reconstruction Problem

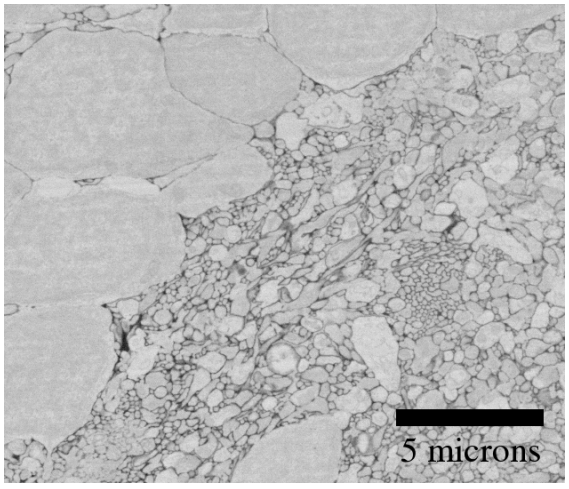
With the significant progress made in the image acquisition process, EM instruments are now able to image tissues at a nanometer resolution and thus produce large-scale data volumes, which require image analysis methods for connectomic reconstruction. The task of reconstruction is to trace individual neuronal processes to the cell body and to identify synaptic connections between neurons. Manual tracing could be a possible solution to reconstruction, but it is impractical. Take an SBFSEM data set as an example. At a resolution of 10 nm $\times$ 10 nm $\times$ 50 nm, sectioning a 200^3 μm^3 sample tissue results in a data set of 1.6×10^{12} voxels [10]. Moreover, enormous numbers of cells in the EM image stacks are densely packed, which can be seen in Figure 1, where two EM (ssTEM and SBFSEM) image stacks and their sample images are shown. Given that it took more than a decade to manually chart the nervous system of the *C. elegans* from a data volume containing approximately $8,000$ serial sections, each of which was cut at 50 nm thickness [22], manual reconstruction of the



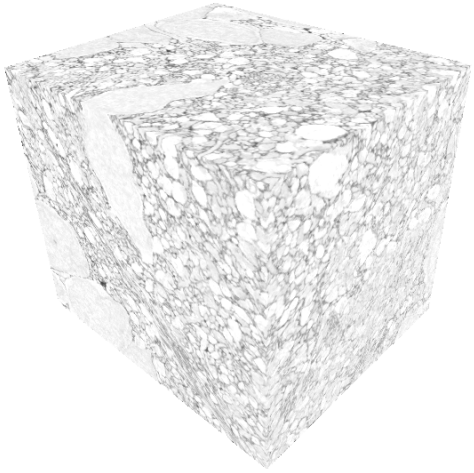
(a) An ssTEM image



(b) An ssTEM image stack



(c) An SBFSEM image



(d) An SBFSEM image stack

Fig. 1. Sample EM images and image stacks. (a) A sample ssTEM image having a size of 512×512 from a data set of Drosophila first instar larva ventral nerve cord (VNC). The resolution is $4 \times 4 \times 50$ nm/pixel. (b) A publicly available ssTEM data set provided by Cardona et al. [24, 25]. (c) A sample SBFSEM image having a size of 631×539 from a data set of larval zebrafish tectum. (b) An SBFSEM image stack. Note that cells in both the ssTEM and the SBFSEM images are densely packed.

sheer volume of data produced by EM techniques appears to be infeasible; therefore, automated (or semi-automated) and reliable reconstruction algorithms for analysis of massive EM volumetric data are necessary.

1. Image Segmentation

Reconstruction involves the tracing of each neurite and the identification of the interconnections. In computer vision, this process is known as image segmentation, partitioning an image into disjoint regions, each of which shares common characteristics. For EM images, the task of segmentation is to group voxels into different sets that represent distinct neurons [26], which is a fundamental step toward a full analysis of neuronal morphological models. Such a task poses unique challenges because the inevitable staining noise, incomplete boundaries, and inhomogeneous staining intensities can increase difficulty of segmentation and subsequent 3D reconstruction and visualization.

To overcome the challenges of analyzing EM images, numerous segmentation approaches have been proposed in the literature. The process of delineating neuron boundaries (membranes) in the EM images can be thought of as boundary detection. Methods to detect neuronal membranes is to extract features that represent the derived membrane information from images and are capable of distinguishing membranes from others. These features include Radon-like features [27] or features obtained by Hessian-based filters [28]. Other approaches include the use of machine learning, where classifiers are trained, such as convolutional networks [29–32], random forests [33, 34], artificial neural networks [35, 28], Adaboost [36], and support vector machine (SVM) [37], to detect neuronal membranes in the EM images. One potential limitation is that these methods require sufficient amount of data along with their corresponding ground truth segmentation for training the classifier to ac-

curately perform the labeling task. That is, creation of labeled data sets can be very laborious. Semi-automated methods that trace neuronal contours through successive cross-sections to build 3D anatomical structures also gained much attention [26, 38–46, 28]. In this category, active contours [41], level-set formulation [38, 39, 47, 48], or watersheds [49, 50] have been used to perform the 2D segmentation. The segmentation work presented in this dissertation is closely related to the semi-automated methods, but it can be easily generalized into a 3D segmentation method.

2. User Correction and Validation

Automated segmentation algorithms for reconstructing the circuitry often involve errors mainly because EM data exhibit high variation in neuronal shapes and intensities, and most of the automated methods are unable to capture all these variations. Those errors require manual correction by using interactive editing software.

Whereas much attempt has been made to design segmentation algorithms, relatively little attention has been given to the development of editing tools [51]. Hence, editing software for segmented EM imaging data remains in its infancy [52], and more effort in this line is needed.

In addition, it is well known that human annotations are intrinsically subjective and biased, and the results of manual correction may vary to some extent. A validation method ensuring that the reconstruction is topologically correct when integrating a set of alternative segmentations is also required [52].

C. Approach

To tackle the unique challenges posed by the massive and complex EM image stacks and to achieve the ultimate goal of reconstruction of neural circuits, a unified frame-

work that includes segmentation algorithms, interactive editing tools, and validation methods adequate for such EM data sets is highly demanded. Thus the goal of this dissertation is three fold:

- **Develop robust segmentation algorithms** for the 3D reconstruction of neuronal processes from stacks of serial EM images in which neurons are tightly packed. These include segmentation methods in both 2D and 3D. The first method segments 2D neuronal contours in the x-y plane by minimizing an energy function into which the segmentation results from adjacent slices are incorporated and then tracks the extracted contours through the cross-sectional plane. Geometrical information of adjacent slices serves as a shape prior and constrains the segmentation process while segmenting one slice. The second method extends the 2D method into a full 3D algorithm by extracting the 3D shape prior directly from the data itself by using the locally symmetric property exhibited in the anatomical structures. These methods provide promising solutions to the reconstruction of neural circuits from EM image stacks.

- **Develop interactive editing tools** that allow the user to quickly correct the erroneous segmentations produced by automated segmentation algorithms in order to obtain morphologically accurate reconstructions. The interactive editing framework provides two stages of editing. First, it gives the user a set of alternative segmentations from which the user can choose the most acceptable one, aiming to minimize the amount of time for manual interaction. Second, similar to the 2D segmentation approach in the aforementioned tracking method, the editing task is formulated as an energy minimization problem. The algorithm takes a number of user inputs, together with the incorrect segmentation and image data constraints, and computes a new segmentation accordingly. These

interactive editing tools provide an efficient way to obtain correct segmentations.

- **Develop validation methods** that estimate the true, unknown segmentation of an EM image when a collection of alternative segmentations, either annotated by humans or produced by segmentation algorithms, are available. This algorithm aims to obtain a topologically correct segmentation whose topology is capable of representing that of the true, unknown segmentation. To this end, it iteratively modifies the topology of the estimated segmentation by means of minimizing the topological disagreements between the estimated segmentation and the provided set of segmentations. This segmentation validation method provides a way to obtain an estimated segmentation that is topologically equivalent and geometrically similar to the true, unknown labeling.

D. Organization of the Dissertation

In this chapter, the motivation for this dissertation work and an overview of the approaches taken in this work were provided. The remainder of the dissertation is organized into 6 chapters.

Chapter II provides a brief background of segmentation algorithms, particularly focusing on Markov random fields and graph cuts, which form a basis of the segmentation algorithms presented in the subsequent chapters. This chapter also provides a review of the approaches for segmentation evaluation that measures the performance of segmentation algorithms and a brief review of surface reconstruction methods that create 3D models from a stack of segmented 2D contours.

In Chapter III, new segmentation methods for segmenting neuronal processes from the stacks of serial EM images are presented. These methods utilize geometrical information of adjacent images and incorporate it into an energy function. Such

geometrical information serves as a constraint in the segmentation process during the minimization of the energy function.

100% accuracy may be hard to achieve for automated segmentation methods because EM images have high variation in neuronal shapes and show low contrast in appearance. These errors require manual correction. Chapter IV presents interactive editing tools that allow the user to refine incorrect segmentations produced by automated segmentation algorithms.

Chapter V presents a validation method that aims to obtain a segmentation that is topologically equivalent and geometrically similar to the true, unknown segmentation given a set of alternative segmentations.

Chapter VI presents the contributions of this dissertation, discusses the potential limitations of the developed methods, and points out future research directions.

Finally, Chapter VII concludes this dissertation.

CHAPTER II

BACKGROUND ON IMAGE SEGMENTATION, SEGMENTATION VALIDATION, AND SURFACE RECONSTRUCTION

With the need of image segmentation in many domains, a large number of segmentation algorithms have been developed in the past decades. Providing a thorough and extensive survey on all segmentation methods is beyond the scope of this dissertation (see [53, 54] for an extensive overview). This chapter, instead, gives a brief background on state-of-the-art image segmentation algorithms, with a particular focus on the use of Markov random fields (MRFs) and graph cuts in segmentation problems.

Aside from providing background on segmentation algorithms, this chapter also reviews evaluation methods that have been generally employed to assess the quality of results produced by segmentation algorithms in the literature. Particular emphasis is placed on the use of evaluation methods that can estimate the ground truth for an image when a set of segmentations of that image are available.

Last, this chapter provides a brief review of surface reconstruction methods that create 3D models from a stack of segmented 2D contours in the parallel plane.

A. Image Segmentation

Image segmentation, that is the partitioning an image into disjoint regions that share common characteristics, is a fundamental step in extracting meaningful (semantic) objects for a higher-level analysis in diverse applications. Prior work on image segmentation algorithms by means of energy minimization can be categorized into continuous and discrete methods based on the spatial domains of the image.

In the continuous domain, parametric active contour models or "snakes" proposed by Kass et al. [55] deform a spline by iteratively minimizing an energy function

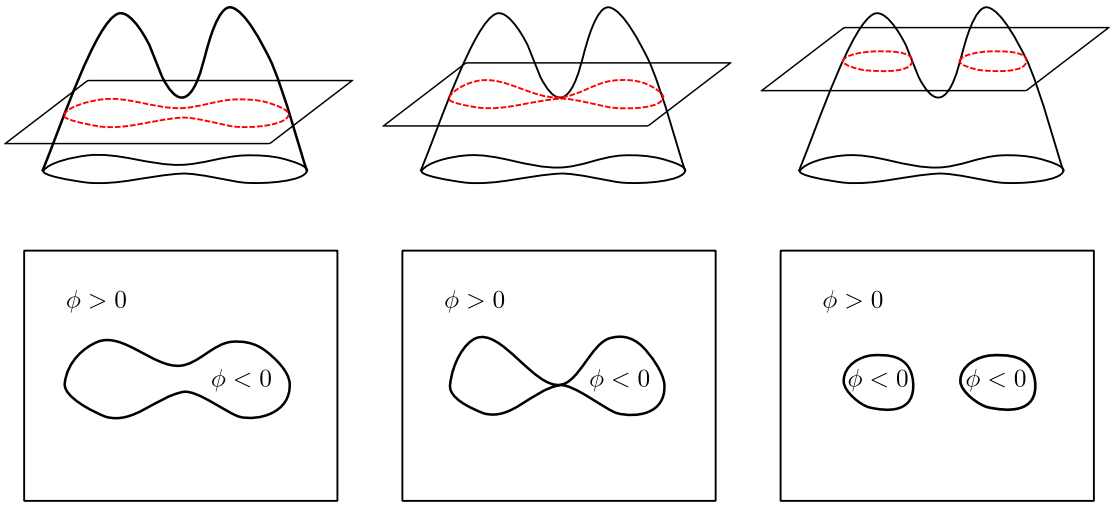


Fig. 2. The level set method. Level set methods embed the interface as the zero level set of the signed distance function ϕ and can handle topological changes, overcoming the limitations of the parametric curve representation.

consisting of internal and external energies. The internal forces are defined over the curvature and contour length, and the external forces are defined based on the image data (e.g. gradient vectors of image intensity [56]). These two forces pull the deformed curve towards the object boundary. However, the parametric models have limitations: they can only represent topologically simple objects, cannot deal with topological changes, and are sensitive to the initial conditions.

The level set method introduced by Osher and Sethian [57], which embeds the interface as the zero level set of the signed distance function, overcomes the limitations of the parametric curve representation. The evolution of the level set function follows the curvature-dependent speed whose equations of motion resemble Hamilton-Jacobi equations. As a result, the level set method can handle topological changes in any dimension, as shown in Figure 2. Because of this property of level set methods, numerous active contour models based on level sets have been developed. According

to the criterion used to guide the curve evolution, they can be divided into two classes: edge-based models [58, 59] and region-based models [60–62]. Edge-based models move the evolving curve towards object boundaries using edge information while region-based models guide the curve evolution based on region intensities. Active contour models based on level sets have several advantages over parametric active contours. However, the level set method is known for its computational demand, and the solution may converge into local minima.

In discrete space, graph-based segmentation represents an image as a graph with graph nodes denoting image pixels or feature points and edges being connections between nodes. In such a setting, image segmentation turns into a graph partition problem or a labeling problem, which is elaborated in the next section. Important tasks in the graph partition include determination of the graph partition cost function and development of efficient optimization algorithms. To this end, the normalized cut (Ncut) algorithm introduced by Shi and Malik [63] uses both the total similarity within groups and the total dissimilarity between different groups as a partition criterion, which is efficiently optimized by solving a generalized eigenvalue problem. One drawback of the normalized cut is that segmenting an image may require huge memory space for storing the matrix used to solve the eigenvalue problem.

Graph-based image segmentation can also incorporate user interaction during the segmentation process. The user interaction has been proved to significantly facilitate in obtaining satisfactory segmentation results. The interactive Random Walker (RW) algorithm proposed by Grady [64] is one example that performs the segmentation upon the user pre-labeled nodes. It determines the probability a random walker starting from an unlabeled pixel reaches to one of the user-labeled pixels, which is equivalent to the solution to the combinatorial Dirichlet problem and is obtained by solving a system of linear equations. Assigning each pixel to the label that is of the

greatest probability gives the final segmentation.

Another interactive segmentation method utilizing user specified pixels to determine the object properties is the graph cuts [65, 66]. The segmentation is formulated as an energy minimization problem built upon the random fields, and the solution to the energy function is obtained by the graph cuts techniques. The detailed discussions on the random fields and graph cuts are provided in the subsequent sections.

1. Markov Random Field Modeling

Since its first introduction by German and German [67] in 1984, Markov Random fields (MRFs) have been used to formulate many vision problems, which can be posed as lableing problems, such as image segmentation [68]. The MRF provides a convenient and consistent framework that uses contextual constraints based on piecewise constancy, which is essential in the image analysis and understanding [68].

a. The Labeling Problem

Image segmentation is considered as a labeling problem that involves assigning a label from a set of labels to each image pixel [68]. More specifically, given a set of sites $\mathcal{P} = \{1, 2, ..., M\}$ and a set of labels $\mathcal{L} = \{l_1, l_2, ..., l_K\}$, the labeling problem is to assign a label f_p from $\mathcal{L}$ to each site $p \in \mathcal{P}$. A possible labeling $f = \{f_1, f_2, ..., f_m\}$ is a mapping from $\mathcal{P}$ to $\mathcal{L}$, that is,

$$f : \mathcal{P} \rightarrow \mathcal{L} \, , \tag{2.1}$$

as shown in Figure 3. According to the random fields theory, the set of sites $\mathcal{P}$ is associated with a random field $F = \{F_1, ..., F_m\}$, where each random variable F_p takes on a value f_p from the set of labels $\mathcal{L}$. The joint event $\{F_1 = f_1, F_2 = f_2, ..., F_M = f_M\}$ is called a *configuration* of F, abbreviated as $F = f$. Given that each site has the

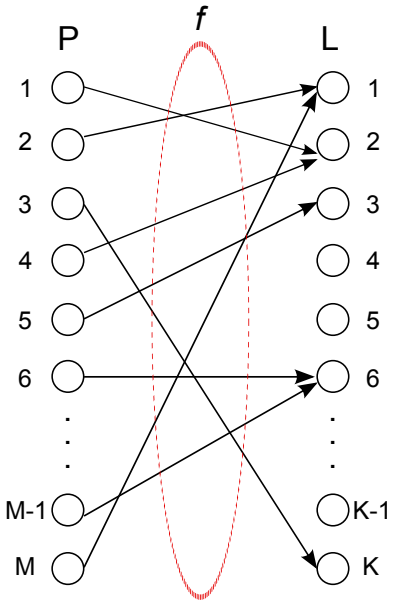


Fig. 3. Illustration of labeling. A labeling f is a mapping from a set of sites $\mathcal{P}$ to the set of labels $\mathcal{L}$. Adapted from [68].

same label set $\mathcal{L}$, the space of all possible labelings, that is the configuration space, is

$$\mathbb{F} = \underbrace{\mathcal{L} \times \mathcal{L} \times \cdots \times \mathcal{L}}_{\text{m times}} = \mathcal{L}^m \ . \tag{2.2}$$

Therefore, the goal of image segmentation is to find an optimal labeling f^*, equivalent to finding the maximum a posteriori (MAP) estimate of the underlying field given the observed image data D. This is

$$f^* = \underset{f \in \mathbb{F}}{\text{argmax}} \Pr\left(f \mid D\right) \ . \tag{2.3}$$

b. Markov Random Fields

According to the Bayes' rule, the posterior probability of Equation 2.3 is given by

$$\Pr\left(f \mid D\right) \propto \Pr\left(D \mid f\right) \Pr\left(f\right) \ , \tag{2.4}$$

where $\Pr(D \mid f)$ is the likelihood of D on f, and $\Pr(f)$ is the prior probability of a particular labeling f, being modeled as a Markov random field (MRF).

An MRF satisfies the following two properties with respect to the neighborhood system $\mathcal{N} = \{\mathcal{N}_p \mid p \in \mathcal{P}\}$:

$$Positivity: \qquad \Pr(f) > 0, \qquad \forall f \in \mathbb{F}\ , \tag{2.5}$$

$$Markovianity: \quad \Pr\left(f_p \mid f_{\mathcal{P}-\{p\}}\right) = \Pr\left(f_p \mid f_{\mathcal{N}_p}\right), \quad \forall p \in \mathcal{P}\ , \tag{2.6}$$

where $\mathcal{P} - \{p\}$ is the set difference, $f_{\mathcal{P}-\{p\}}$ is the set of labels of sites in $\mathcal{P} - \{p\}$, $\mathcal{N}_p$ is the neighboring sites of p, and $f_{\mathcal{N}_p}$ denotes the set of labels of the neighboring sites of p.

Furthermore, according to Hammersley-Clifford theorem [69], a random field with Markov property obeys a Gibbs distribution, which takes the following form:

$$\Pr(f) = \frac{1}{Z}\exp\left(-E(f)\right)\ , \tag{2.7}$$

where Z is a normalizing constant called the partition function that is defined as

$$Z = \sum_{f \in \mathbb{F}} \exp\left(-E(f)\right)\ , \tag{2.8}$$

and $E(f)$ is the Gibbs energy function, which is

$$E(f) = \sum_{c \in \mathcal{C}} V_c(f_c)\ , \tag{2.9}$$

where $\mathcal{C}$ is the set of cliques, c is a clique defined as a subset of sites in $\mathcal{P}$, and $V_c(f_c)$ is a clique potential.

Taking a log likelihood of Equation 2.4, the MAP estimate of $\Pr(f \mid D)$ is equiv-

alent to minimizing the following energy function:

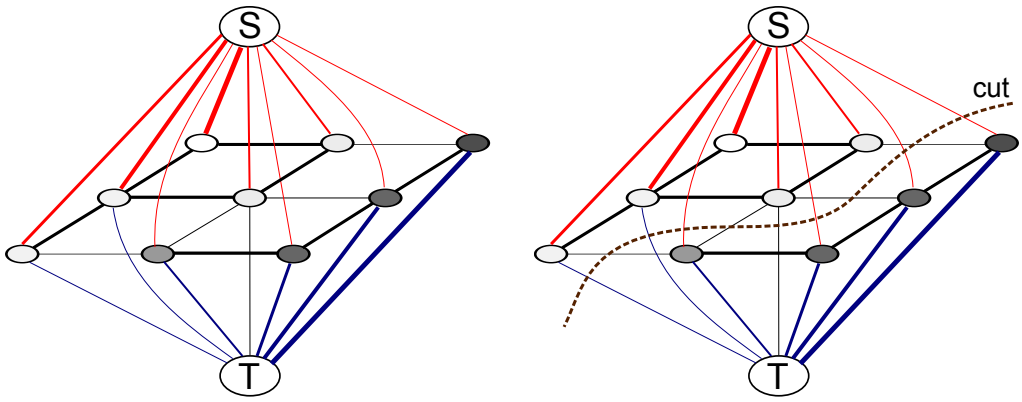
$$-\log \Pr\left(f \mid D\right) = E\left(f \mid D\right) = \sum_{p \in \mathcal{P}} V_p\left(f_p \mid D\right) + \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p} V_{pq}\left(f_p, f_q \mid D\right) \ , \qquad (2.10)$$

where $V_p\left(f_p \mid D\right)$ and $V_{pq}\left(f_p, f_q \mid D\right)$ are the unary and piecewise clique potentials, respectively. More specifically, the unary potential $V_p\left(f_p \mid D\right)$ measures the cost of assigning node p a label f_p. The piecewise potential $V_{pq}\left(f_p, f_q \mid D\right)$ represents the cost of assigning labels f_p and f_q to nodes p and q, respectively, which imposes the spatial smoothness using the Potts model [70].

2. Graph Cuts

Several algorithms have been proposed to minimize the energy function shown in Equation 2.10: the Iterated Conditional Modes (ICM) [71], the belief propagation algorithm (BP) [72], the tree-reweighted message passing algorithm (TRW) [73], and graph cuts [65, 66]. This section focuses on the discussion on the graph cut algorithm.

Graph cuts represent an image as a weighted directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with a set of vertices (nodes) $\mathcal{V}$ representing pixels or image regions and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ connecting the nodes. Each edge is associated with a nonnegative weight. The set $\mathcal{V}$ includes the nodes of the set of image pixels $\mathcal{P}$ and two additional nodes, the source s and the sink t. All nodes $p \in \mathcal{V}$ are linked to the terminals s and t with weight w_{sp} and w_{pt}, respectively. Node p is also connected to its neighboring nodes q, based on a specified neighborhood system, with weight w_{pq}. Figure 4(a) shows a graph construction example for a 3×3 image, where each node is connected to its neighboring nodes by a 4-neighborhood system. Edges between the nodes and the terminals are called t-links, and edges between a node and its neighboring nodes are called n-links.



(a) Graph construction (b) A cut

Fig. 4. Graph construction and an associated cut. (a) shows the graph construction for a 3×3 image. The thickness of lines represents the weighting of edge weights. The thicker the lines are, the more weight they carry. (b) shows the minimum cost cut that separates two terminals as well as partitions the remaining nodes into two disjoint groups, and a segmentation of the image is hence obtained. Adapted from [65].

A cut $\mathcal{E}_c \subset \mathcal{E}$, also referred to as an *st*-cut, is a subset of edges $\mathcal{E}$ that separates terminals in the induced graph $\mathcal{G}' = \langle \mathcal{V}, \mathcal{E} \setminus \mathcal{E}_c \rangle$. Removing edges in the cut $\mathcal{E}_c$ partitions the nodes into two disjoint subsets, S and T, such that $s \in S$ and $t \in T$, as illustrated in Figure 4(b). The cost of a cut, denoted as $|\mathcal{E}_c|$, is the sum of costs of all edges in $\mathcal{E}_c$, that is

$$|\mathcal{E}_c| = \sum_{u \in S, v \in T, (u,v) \in \mathcal{E}} w_{uv} \ , \tag{2.11}$$

where u and v are nodes in the $\mathcal{V}$, and w_{uv} is the weight associated with edge (u, v). Because a graph represents an image, the partitioning of a graph by a cut corresponds to a segmentation of the image. Each different cut has an associated cost. Therefore, image segmentation problem then turns into finding a minimum cost cut that best partitions the graph. Algorithms to obtain the minimum cost cut via the calculation

of the maximum flow based on the theorem of Ford and Fulkerson [74–76] are discussed below.

a. Energy Minimization Using Graph Cuts

Minimizing the energy $E(f \mid D)$ in Equation 2.10 is difficult because it is a nonconvex function. The development of graph cuts provides an efficient way to minimize this energy function. In order for the graph cuts to minimize it, the energy function $E(f \mid D)$ should be in a specialized class such that the minimum cost cut on the constructed graph also minimizes the energy. Shown by Kolmogorov and Zabih [70], a characterization of functions in this class that can be minimized by graph cuts is *submodular*. Before any further discussion on submodular functions, an introduction to the *regular functions*, which have an interesting relation to the submodular functions, is first presented.

b. Regular Functions and Submodular Functions

Following the definition in [70], a *regular function* is a function of n binary variables, denoted as $E(x_1, ..., x_n)$. In order to define the regularity of arbitrary functions of binary variables, it requires to define the *projections* of a function.

Definition 1. *Let $E(x_1, ..., x_n)$ be a function of n binary variables, and let I, J be a disjoint partition of the set of indices $\{1, ..., n\}$: $I = \{i(1), ..., i(m)\}, J = \{j(1), ..., j(n-m)\}$. Let $\alpha_{i(1)}, ..., \alpha_{i(m)}$ be binary constants. A projection $E' = E\left[x_{i(1)} = \alpha_{i(1)}, ..., x_{i(m)} = \alpha_{i(m)}\right]$ is a function of $n - m$ variables defined by*

$$E'\left(x_{j(1)}, ..., x_{j(n-m)}\right) = E(x_1, ..., x_n) \ ,$$

where $x_i = \alpha_i$ for $i \in I$, and the variables $x_{i(1)}, ..., x_{i(m)}$ are fixed.

Using the above definition, regular functions are defined as below.

Definition 2.

- *All functions of one variable are regular.*
- *A function E of two variables is called regular if*

$$E(0,0) + E(1,1) \leq E(0,1) + E(1,0) \ .$$

- *A function E of more than two variables is called regular if all projections of E onto two variables are regular.*

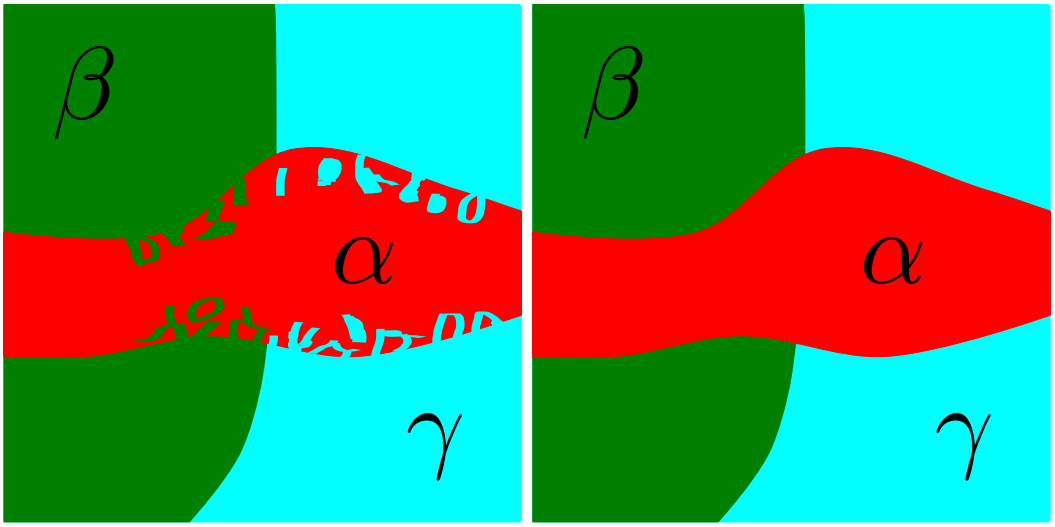
The relation between regular functions and submodular functions is established through the following definition of submodularity of a set function.

Definition 3. *Let S be a finite set and $g : 2^S \rightarrow \mathbb{R}$ be a real-valued function defined on the set of all subsets of S. g is called submodular if, for all $X \subset S$ and $i, j \in S - X$, it satisfies*

$$g(X \cup \{j\}) - g(X) \geq g(X \cup \{i,j\}) - g(X \cup \{i\}) \ .$$

Based on the above definition, submodularity reduces to the definition of regularity; submodular functions are thus the same as regular functions [70].

Submodularity is related to the *smoothness* property in the labeling problem. Using the Potts model to impose the spatial smoothness in $V_{pq}(f_p, f_q \mid D)$ in Equation 2.10 ensures a labeling f is smooth in the homogeneous regions and has different labels of adjacent pixels at object boundaries. Such an energy function is called *discontinuity preserving*. Global minimization of the discontinuity-preserving energy functions is NP-hard, so researchers focus on the development of efficient approximate solutions. An efficient algorithm that minimizes such energy functions is the expansion move algorithm proposed by Boykov et al. [74] based on graph cuts. Given a



(a) Initial labeling (b) α-expansion

Fig. 5. An example of α-expansion. (a) The initial labeling of an image. (b) The labeling after α-expansion is performed. As can be seen, a large number of pixels in (b) change their labels to α. Adapted from [74].

labeling f and a label α, a new labeling f' is called an *α-expansion* from f if $f_\alpha \subset f'_\alpha$ and $f'_l \subset f_l$ for any label $l \neq \alpha$, meaning that a set of pixels change their labels to α, as illustrated in Figure 5. The algorithm starts from an arbitrary labeling and iteratively finds a new labeling when expansion moves are allowed, that is, the energy of the new labeling is lower than that of the current labeling. If there is no *α-expansion* moves, the algorithm terminates, thus giving a labeling with the lowest energy with respect to expansion moves. The essential step in the expansion algorithm is to find the optimal *α-expansion* moves, which can be efficiently obtained by using graph cuts that utilize the maximum flow and minimum cut algorithm.

c. Maximum Flow and Minimum Cut

The max-flow min-cut theorem by Ford and Fulkerson [75, 76] states that given a capacitated network, the maximum flow from the source s to the sink t is equal to the capacity of the minimum cut. To find a maximum flow in a given directed graph, the Ford-Fulkerson algorithm repeatedly finds an augmenting path from the source to the sink and increases the flow by determining the excess flow capacity of this path. The maximum flow is determined if and only if there is no augmenting paths, hence obtaining the minimum cost cut that separates the source s and the sink t and partitions the nodes into two subsets. The time complexity of the Ford-Fulkerson algorithm is $O\left(|\mathcal{V}||\mathcal{E}|^2\right)$.

B. Segmentation Validation

Extensive development of segmentation methods raises the need for performance evaluation. Segmentation evaluation, also known as segmentation validation in medical imaging or biomedical imaging, qualitatively and quantitatively analyzes segmentation algorithms' outputs and thus provides a measure of their performance. This section provides a review on segmentation evaluation.

1. Supervised and Unsupervised Evaluation

Segmentation evaluation can be either subjective or objective. The most common method of subjective segmentation evaluation is that a human expert visually compares the quality of segmentation results of different segmentation algorithms, which is a tedious and time-consuming task. Besides, visual comparison also limits the evaluation to a small number of segmentation results. Another alternative is objective evaluation, which can be classified into two categories, *unsupervised evaluation* and

supervised evaluation, depending on whether a ground truth segmentation is used or not.

Supervised evaluation methods assess the quality of a segmentation algorithm by comparing its results against a reference ground truth (i.e. a manually segmented reference image) using segmentation evaluation metrics. Commonly used evaluation metrics for evaluating the object accuracy are Jaccard index [77], fuzzy Jaccard index [77], Dice similarity coefficient (DSC) [78], F-measure (i.e. the weighted harmonic mean of precision and recall), and so on. Those metrics quantitatively measure the segmentation methods' performance by using the amount of overlap between their results and the reference images as a measure of the degree of segmentation similarity. For assessing the boundary accuracy, the Hausdorff distance, a measure of the largest distance between two contours, is a commonly used metric. The advantage of supervised evaluation methods is that direct comparison between a machine-generated segmentation and a human-segmented reference provides a finer resolution of evaluation; however, the creation of manual segmentations is subjective, arduous, and time-consuming [79].

Conversely, unsupervised evaluation methods assess the quality of segmentation by measuring how well the segmentation results meet a set of criteria that are defined based on principles of visual perception, for example, Gestalt laws of perception (law of closure, similarity, proximity, symmetry, and continuity). As a result, it is not necessary to compare the segmentation results against the ground truth segmentation. The criteria of a good segmentation are [79]:

- Regions should be uniform and homogeneous with respect to some characteristic(s).
- Adjacent regions should have significant differences with respect to the charac-

teristic on which they are uniform.

- Region interiors should be simple and without holes.
- Boundaries should be simple, not ragged, and be spatially accurate.

Given the above criteria, metrics for unsupervised evaluation methods aim to measure inter-region disparity and intra-region uniformity based on color, texture, entropy, or normalized mutual information [80] or to evaluate the shape of an object. In addition to this, unsupervised segmentation evaluation can also be considered as the computation of the likelihood of a pixel or a region belonging to the foreground. The computation of the likelihood of label assignments can be associated with uncertainty measure of solutions produced by inference algorithms, for instance, dynamic graph cuts [81].

2. Methods for Generation of Ground Truth for Supervised Evaluation

In medical image segmentation evaluation, supervised evaluation methods are preferred. Because medical images have low contrast and ambiguous boundaries, an unsupervised evaluation metric meeting the criteria of a good segmentation is hard to define. For supervised evaluation, it requires the creation of ground truth annotations, where human experts manually delineate the object boundary. It is well known that human annotators are intrinsically biased and subjective so manual segmentations sometimes vary significantly. Humans not always agreeing on the object boundary brings the problem of boundary discrepancies in the manual segmentations. A potential solution to this problem is using a combination of multiple manual annotations to represent the true segmentation.

To find the true segmentation of multiple manual segmentations, label fusion methods are used. Commonly used methods for label fusion are majority voting,

global weighted voting, and local weighted voting.

Majority voting is a widely used strategy for the label fusion because of its simplicity. It treats each individual segmentation equally and assigns a pixel the label the most segmentations agree on. Mathematically, let $y^1, ..., y^N$ be the N binary segmentations and $\hat{y}$ be the true, unknown segmentation to be estimated. The estimated true segmentation $\hat{y}_i$ at point i is given as

$$\hat{y}_i = \begin{cases} 1 & \text{if } \frac{1}{N}\sum_{j=1}^{N} y_i^j \geq 0.5 \\ 0 & \text{otherwise} \end{cases} . \tag{2.12}$$

Another strategy for the label fusion is global weighted voting that weights each segmentation differently according to the performance of each evaluator. Simultaneous truth and performance level estimation (STAPLE) algorithm proposed by Warfield et al. [82] belongs to this category. STAPLE uses an iterative expectation-maximization algorithm to measure the performance of experts and estimates the underlying true segmentation by optimally combining each segmentation depending on each expert's performance level. STAPLE measures two performance level parameters of each expert: (1) sensitivity (i.e. true positive rate), the probability of an annotator labeling a pixel as foreground if the true label is foreground; and (2) specificity (i.e. 1-false positive rate), the probability of an annotator labeling a pixel as background if the true label is background. In sum, the STAPLE algorithm iterates between the estimation of conditional expectation of the complete data log likelihood given the current estimates of the expert parameters (i.e. E-step) and the estimation of the performance parameters by maximization (i.e. M-step). In STAPLE, The convergence to a local optimum is guaranteed.

In addition to its usefulness on the segmentation validation, the STAPLE algorithm has also been applied to atlas construction [83]. Recognizing that the accuracy

of combined multiple classifiers is significantly higher than that of an individual classifier, Rohlfing et al. [84] extended binary segmentation combination to multi-class combination. Commowick and Warfield [85] proposed another extension of STAPLE, in which prior probabilities for the expert performance parameters are incorporated to avoid convergence to undesirable local optima while dealing with partial delineations.

In contrast to giving the same weighting to each pixel in a segmentation, local weighted voting methods [86, 87] assign each pixel a different weight according to a local estimation of the segmentation performance. The estimated segmentation $\hat{y}_i$ at point i is defined as

$$\hat{y}_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{N} p_i^j y_i^j \geq 0.5 \\ 0 & \text{otherwise} \end{cases}, \tag{2.13}$$

where p_i^j denotes the local weighting associated at point i, which is a similarity measure estimated from a local neighborhood center at point i.

C. Surface Reconstruction of Neuronal Processes

In order to visualize 3D models of neurons and to extract geometrical information, we need to reconstruct a 3D surface from a stack of segmented 2D neuronal contours in the x-y plane. Prior work on 3D reconstruction from 2D contours can be divided into two categories: volumetric and contour stitching methods.

The volumetric method reconstructs 3D surface by interpolating the 2D cross-sectional contours in the z direction. The signed distance function is usually used to represent the 2D contour, and intermediate implicit functions are generated by linearly interpolating these 2D contours [88]. The 3D model of an object is reconstructed by extracting the zero iso-contours from the interpolated functions. An alternative approach for interpolation of adjacent contours is by morphing one contour into the

next [89], which propagates the input contours as 2D level sets using continuous speed functions.

The contour stitching method for surface reconstruction focuses on connecting the vertices of adjacent contours in order to form a mesh. The seminal work in this category is that of Fuchs et al. [90], which creates the reconstructed surface by finding minimum cost paths in a directed toroidal graph. Another method proposed by Barequet et al. [91] uses the information from neighboring layers to guide the intepolation of the current layer, thus creating a smoother surface. Bajaj et al. [92] proposed a unified framework to address three major problems in the reconstruction: the correspondence (how to find correct connections between the contours of adjacent slices), tiling (how to use slice chords to triangulate the strip lying between contours of two adjacent slices into tiling triangles), and branching (how to deal with a contour in one slice corresponding to more than one contour in an adjacent slice) problems. To this end, they imposed three constraints on the reconstruction surface and then derived correspondence and tiling rules.

The contour stitching method has also been applied to reconstruction of 3D models from segmented 2D neuronal membranes of ssTEM images by Bajaj and Gillette [93] and Edwards and Bajaj [94]. They dealt with the problem of intersections between neurons, which occurs because of densely packed branching structures in neurons, while multiple neurons are visualized in the same model.

D. Summary

This chapter reviewed the theoretical fundamentals of graph cuts, which serve as a basis for the development of segmentation methods and interactive editing tools. This chapter also provided a review on validation methods, with a focus on approaches

to label fusion used for the estimation of ground truth given a set of segmentations. A limitation of these fusion methods is that the fusion is performed at a pixel level, not taking into account the topology of the segmented regions. This motivates the development of a fusion method ensuring that a segmentation is topologically correct, which is presented in Chapter V. Finally, a review of surface reconstruction methods for generating 3D models from a stack of 2D contours was also provided.

CHAPTER III

IMAGE SEGMENTATION USING GRAPH CUTS

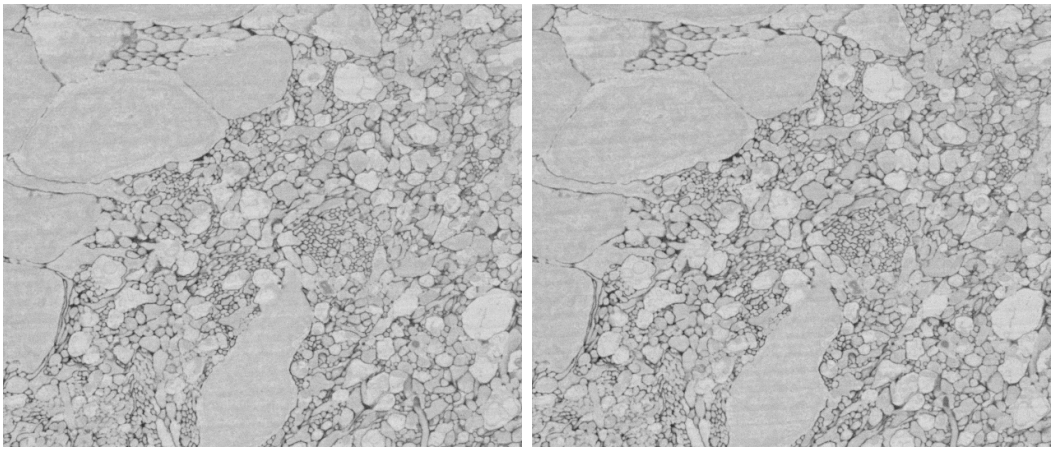
This chapter presents two segmentation approaches for the reconstruction of neural circuits from the stacks of serial EM images in detail. First, a 2D segmentation method is described. This algorithm segments neuronal contours in the x-y plane and tracks them through the cross-sectional plane. Second, the approach to performing a 3D segmentation in EM images is presented, which is a natural extension of the previous 2D segmentation method.

A. 2D Contour Tracking and Segmentation Using Graph Cuts

This section details a tracking and segmentation framework that performs segmentation of neuronal contours in the x-y plane and tracks them through the z direction. While segmenting neuronal contours in the 2D plane, this framework employs the segmentation results from the previous image, represented by a distance function, to overcome the problem of boundary ambiguity. Utilizing such geometrical information (i.e. shape information) available from adjacent images greatly helps correctly determine the boundaries of cells because the cells' shapes between adjacent images do not vary drastically; instead, their shapes are similar, as shown in Figure 6.

Recall from the previous chapter that given a set of sites $\mathcal{P} = \{1, 2, ..., M\}$ and a set of labels $\mathcal{L} = \{l_1, l_2, ..., l_K\}$, the segmentation is to assign a label f_p from $\mathcal{L}$ to each site $p \in \mathcal{P}$. The optimal segmentation is to find a labeling f that minimizes the energy function:

$$E\left(f \mid D\right) = \sum_{p \in \mathcal{P}} V_p\left(f_p \mid D\right) + \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p} V_{pq}\left(f_p, f_q \mid D\right) \ , \tag{3.1}$$



(a) (b)

Fig. 6. Two adjacent slices from an SBFSEM image stack. Cells' shapes between adjacent slices do not vary drastically, but are similar.

where $V_p(f_p \mid D)$ and $V_{pq}(f_p, f_q \mid D)$ are the unary and piecewise clique potentials, respectively. The unary potential $V_p(f_p \mid D)$ measures the cost of assigning node p a label f_p. The piecewise potential $V_{pq}(f_p, f_q \mid D)$ represents the cost of assigning labels f_p and f_q to nodes p and q, respectively. In the proposed segmentation method, the flux of image gradients and geometrical information of adjacent images constitute the unary potential, and the image intensity the piecewise potential.

1. Unary Potential: Flux and Shape Prior

This section elaborates on the method of calculating the flux of image gradients and of representing the shape information of adjacent images. It also details the approach for integrating these two terms.

a. Flux

Flux has recently been introduced by Vasilevskiy and Siddiqi [95] into image analysis and computer vision. They incorporated flux into a level-set method to segment blood vessel images. After that, flux has also been integrated into graph cuts [96, 97] to improve the segmentation accuracy. The introduction of flux into graph cuts can reduce the discretization artifacts that is a major shortcoming in graph cuts [96]. By definition, considering a vector field v defined for each point in $\mathcal{R}^3$, the total inward flux of the vector field through a given continuous hypersurface S is given by the surface integral [95]

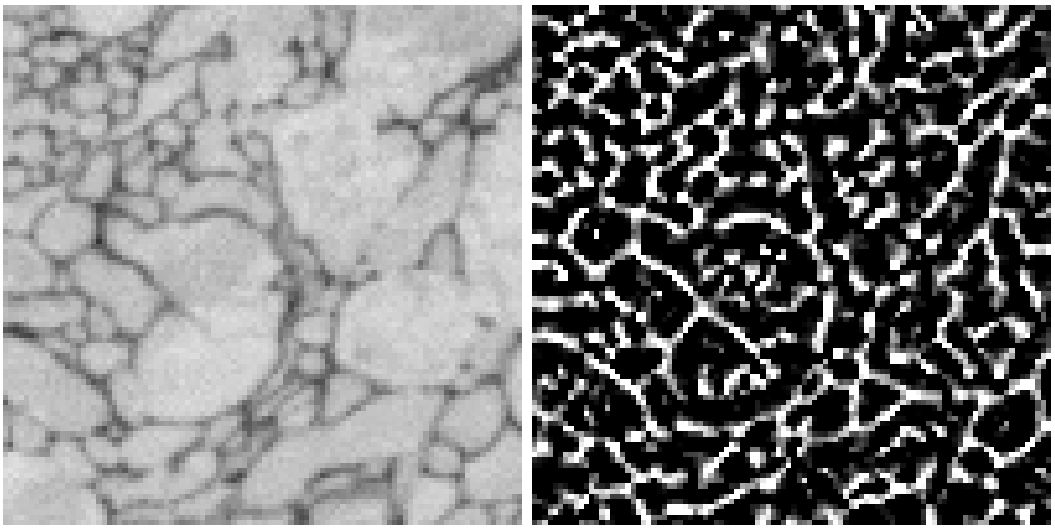
$$F(S) = \int_S \langle N, v \rangle \, dS \ , \tag{3.2}$$

where $\langle , \rangle$ is the Euclidean dot product, N are unit normals to surface element dS consistent with a given orientation. Inward and outward are two possible orientations that can be assigned to S. In the implementation of Equation 3.2, calculation of the flux is simplified by utilizing the divergence theorem that states that the integral of the divergence of a vector field v inside a region equals to the outward flux through a bounding surface. The divergence theorem is given by

$$\int_R \text{div v} \ dR = \oint_S \langle N, v \rangle \, dS \ , \tag{3.3}$$

where R is the region. For the numerical implementations, the flux is thought of passing through a sphere in the case of 3D. As a consequence, N is the outward normal at each point on the contour, and v is defined as the normalized (unit) image gradient vector field of the Gaussian smoothed image I_σ, that is,

$$v = \frac{\bigtriangledown I_\sigma}{\| \bigtriangledown I_\sigma \|} \ , \tag{3.4}$$



(a) SBFSEM image (b) Associated flux of (a)

Fig. 7. Flux of the gradient vector fields of an SBFSEM slice. (a) shows part of an original gray-scale intensity image from the SBFSEM stack. (b) is the associated flux of image gradients of (a), where the foreground objects have negative flux (dark), and the background objects have positive flux (bright).

where σ is the standard deviation of a Gaussian kernel. The flux at a point p is given by [98]

$$F(p) = \sum_{q \in \mathcal{N}_p} \langle N_q, v_q \rangle \ , \tag{3.5}$$

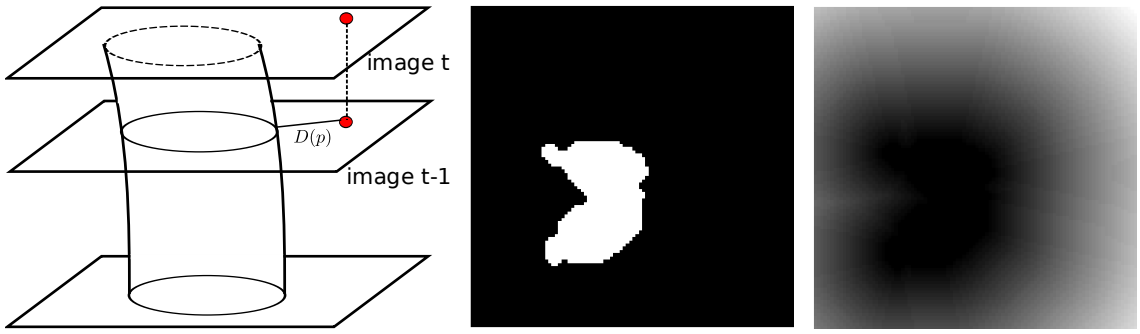
where $\mathcal{N}_p$ indicates the neighbor points of a point p, and N_q and v_q are the unit outward normal and the image gradient at point q, respectively. In this framework, the flux is computed in 2D. Figure 7(b) shows the flux of gradient vector fields of Figure 7(a). The foreground object has negative flux (dark) whereas the background has positive flux (bright).

b. Distance Function

Because the neuron appearances between adjacent images are similar, use of the shape information from the previous segmentation can greatly help determine the correct neuron boundary if missing or blurred boundaries are encountered. The incorporation of this geometrical information into the energy function constrains the segmentation of a slice, aiming to obtain a segmentation similar to those of adjacent images. The geometrical information is represented by a distance function. Let O_{t-1} denote the object in image $t-1$. The distance function $D(p)$ at a point p in image t is given by

$$D(p) = \begin{cases} \|p - o_p\| & p \text{ is outside } O_{t-1} \\ 0 & p \text{ is inside } O_{t-1} \end{cases}, \tag{3.6}$$

where $\|p - o_p\|$ represents the Euclidean distance from p to the nearest object pixel $o_p \in O_{t-1}$. Figure 8(a) depicts the method of computing a distance function utilizing the segmentation result of image $t-1$ shown in Figure 8(b), and Figure 8(c) shows the resulting distance function of a segmentation. The distance function penalizes pixels outside the previously segmented objects, but pixels inside the previously segmented objects get no penalty. In other words, the larger the distances between the pixels outside the previously segmented objects and previously segmented objects' boundaries, the lower the possibility of those pixels belonging to the foreground. One may argue that the choice of the above distance function is not symmetric for the foreground and the background. Another symmetric distance function [99] to the foreground and the background was also implemented and tested; however, the asymmetric and symmetric distance functions both yielded similar segmentation results.



(a) (b) (c)

Fig. 8. Illustration of how a distance function is computed and the resulting distance function. (a) An illustration of how to compute the distance function. (b) The segmentation result of image $t-1$. (c) The resulting distance function in which dark represents the segmentation at image $t-1$, and bright indicates the points outside the previously segmented object. This distance function penalizes pixels outside the previously segmented objects but pixels inside the previously segmented objects get no penalty.

c. Incorporating Flux and Distance Function

Combining the flux of gradient vector fields and the geometrical information of adjacent images represented as a distance function yields a new unary term. By utilizing the method mentioned in [97] of how to set the flux in the t-links, edge weights between node p and terminals s and t that incorporate the flux and the distance function are defined as

$$\begin{aligned} w_{sp} &= -\min\left(0, F\left(p\right)\right), \\ w_{pt} &= \max\left(0, F\left(p\right)\right) + \alpha D\left(p\right), \end{aligned} \tag{3.7}$$

where $F\left(p\right)$ denotes the flux at point p, and α is a positive parameter adjusting the relative importance of the shape prior $D\left(p\right)$. In most cases, the weighting parameter α is empirically set to a fixed value yielding the best results, but given the high variation in illumination of EM images, the value of α is automatically tuned based

on an image' content. The approach for dynamically adjusting this parameter is detailed in the next section.

d. Adaptive Weighting of Shape Prior by Curvedness

The weighting parameter α in Equation 3.7 controls the effect of the segmentations from adjacent slices, and this influence is automatically adjusted based on the spatially varying image data. One principle of deciding the level of effect of the shape prior depends on the strength of edge evidence. That is, little influence of shape prior is given at locations where demonstrate strong edge evidence. On the other hand, more weight is assigned to regions with low edge evidence. One way to achieve this principle is through the use of local curvature information.

The curvature is calculated as follows. Let the first-order gauge coordinates as directions v (isophote tagent) and w (isophote normal), given by

$$\hat{\omega} = \frac{1}{\sqrt{L_x^2 + L_y^2}} \begin{bmatrix} L_x \\ L_y \end{bmatrix} \tag{3.8}$$

and

$$\hat{v} = \hat{\omega}_{\perp} \ , \tag{3.9}$$

where L_x and L_y denotes the first derivatives (i.e. the gradient) of an image's luminance function $L(x, y)$ (i.e. pixel values) with respect to the x and y direction, respectively. In this coordinate, the isophote curvature κ is defined as the rate of change w'' of the tangent vector w'

$$\kappa = -\frac{L_{vv}}{L_w} \ . \tag{3.10}$$

In Cartesian coordinates, it becomes [100]

$$\kappa = -\frac{L_{vv}}{L_w} = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{\left(L_x^2 + L_y^2\right)^{3/2}} \,, \tag{3.11}$$

where L_{xx} is the second derivative with respect to x, L_{yy} is the second derivative with respect to y, and L_{xy} is the mixed partial second derivative in the x and y directions.

Here, the regularization of the relative weighting of the parameter α depends on the curvedness [101, 102], which indicates the amount of deviation from flatness and is defined as

$$C = \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2} \,. \tag{3.12}$$

Note that the curvedness is low at flat areas and high around the edges of an object. The curvedness is normalized to $[0, 1]$ before used to calculate the weight. To give little effect of shape prior at locations demonstrating strong edge evidence and more effect at flat areas, the weight that controls the influence of the segmentation results from the adjacent image at point p is given as

$$\alpha\left(p\right) = \exp\left(-C\left(p\right)\right). \tag{3.13}$$

This is, when the value of curvedness is close to 1, the weight is close to 0.

2. Piecewise Potential: Image Intensity

In the EM images, the foreground and background can be discriminated by their gray-scale intensities. Compared to the background, the foreground objects usually have higher intensity values. Boundaries can thus be determined if the intensity differences between points are large. To capture the boundary discontinuity between pixels, the

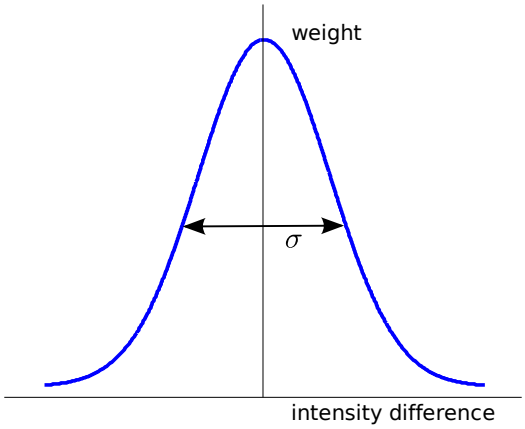


Fig. 9. Illustration of weight settings for piecewise potentials. The x and y axes represent the intensity difference and weight, respectively. σ is the standard deviation of the Gaussian function.

weight between node p and its neighbor q is defined as [66]

$$w_{pq} = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{\|p - q\|}, \tag{3.14}$$

where I_p and I_q are point intensities ranging from 0 to 255, $\|p - q\|$ is the Euclidean distance between p and q, and σ is a positive parameter set to 30. The first term in the right hand side of Equation 3.14 is a Gaussian function that is illustrated in Figure 9. Hence, Equation 3.14 penalizes a lot for edges with similar gray-scale intensities while it penalizes less for those with larger gray-scale differences. In other words, a cut is more likely to occur at the boundary, where the edge weights are small. For 2D segmentation, an 8-neighborhood system is used.

3. Experimental Results

Experiments were conducted on synthetic data sets and stacks of serial EM images in order to evaluate the performance of the proposed approach. Before the segmentation

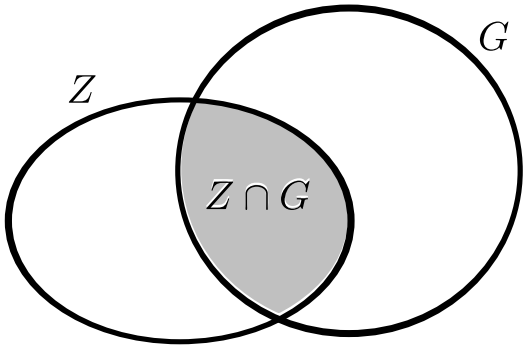


Fig. 10. Illustration of how Dice similarity coefficient (DSC) is computed given two segmentations. DSC is calculated as $\frac{2|Z \cap G|}{|Z|+|G|}$, where Z and G are two segmentations, $|Z \cap G|$ is the common volume between Z and G, and $|\cdot|$ denotes the number of voxels.

results are shown, the evaluation metrics used for measuring the performance of a segmentation algorithm is first introduced in the next section.

a. Segmentation Evaluation Metrics

Dice similarity coefficient (DSC) [78] and F-measure are two evaluation metrics that are commonly used to evaluate how good a segmentation algorithm is compared to a human annotation. Let Z be the set of voxels of the segmentation results produced by a segmentation algorithm and G be the ground truth annotated by a human, the definition of these two metrics is given as follows.

- **Dice Similarity Coefficient (DSC)** measures the amount of overlap between the obtained segmentation results and the ground truth, defined as

$$\mathrm{DSC} = \frac{2|Z \cap G|}{|Z| + |G|}\,, \tag{3.15}$$

  where $|Z \cap G|$ is the common volume between Z and G, and $|\cdot|$ is the number of voxels. 0 indicates no overlap between two segmentations, and 1 means two

Table I. Quantitative results of SBFSEM data from Andres et al. [33]. A classifier was trained for neuronal membrane segmentation, and the performance of the classifier was evaluated on the training data and testing data. According to their results, the acceptable values for DSC, precision, recall, and F-measure are around 0.8 for SBFSEM data (testing data), and values above 0.87 are considered good (training data).

	DSC	Precision	Recall	F-measure
Training	0.8770	0.8714	0.8825	0.8769
Testing	0.7788	0.7567	0.8021	0.7787

segmentations are identical. Figure 10 illustrates how DSC is computed given two segmentations.

- **F-measure** is the weighted harmonic mean of precision and recall, defined as

$$F = \frac{2PR}{P+R}\,, \tag{3.16}$$

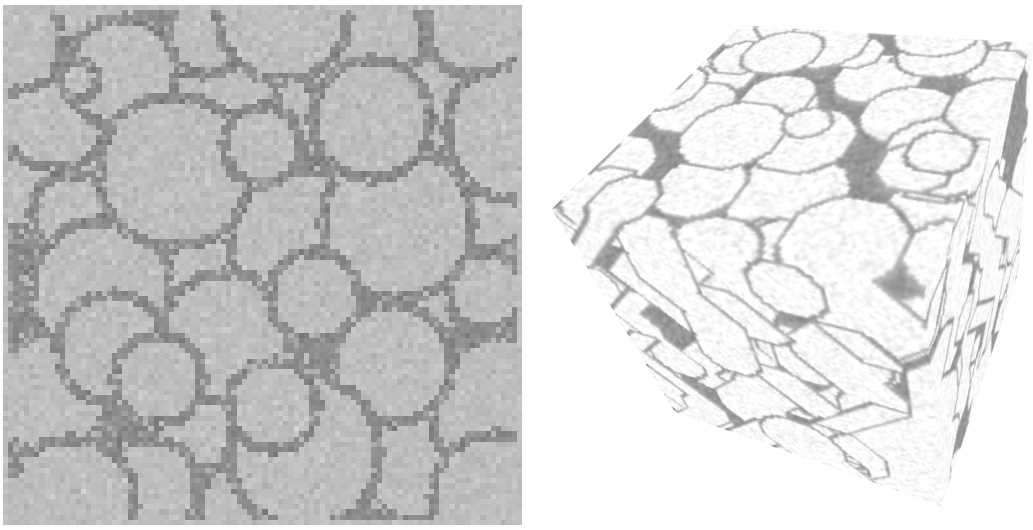
where P and R are the precision and recall of the segmentation results relative to the ground truth, which are given as

$$P = \frac{|Z \cap G|}{|Z|} \tag{3.17}$$

and

$$R = \frac{|Z \cap G|}{|G|}\,. \tag{3.18}$$

Table I shows the quantitative results from SBFSEM data in Andres et al. [33], in which a classifier was trained for neuronal membrane segmentation. According to their results, the acceptable values for DSC, precision, recall, and F-measure are around 0.8 for SBFSEM data (testing data), and values above 0.87 are considered



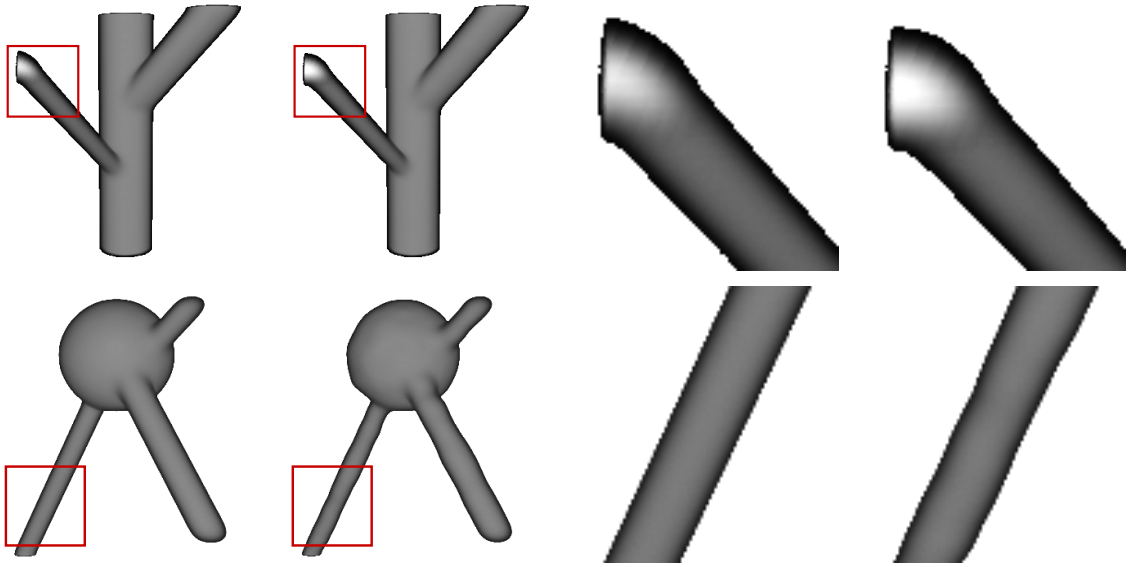
(a) An image from (b) (b) A synthetic image stack

Fig. 11. A synthetic image and a synthetic data set. (a) is a noisy image with 100×100 pixels selected from the synthetic image stack in (b). (b) shows one of the two synthetic image stacks, which contains 100 images.

good (training data).

b. Synthetic Data

Two image stacks (data set 1 and data set 2) were generated, each having the size of $100 \times 100 \times 100$. Gaussian noise was added to each image slice to simulate the noise obtained during the image acquisition process. Three different levels of Gaussian noise with standard deviation $\sigma = 0.0447, 0.0632$, and 0.0775 were added to the two synthetic data sets, thus resulting in a total of 6 image stacks. Shown in Figure 11(b) is one of the synthetic image stacks, and in Figure 11(a) is a noisy image from the synthetic image stack in Figure 11(b). The reconstruction results of the two synthetic image stacks are shown in Figure 12(b), and their ground truth is given in Figure 12(a) accordingly. As can be seen from the close-up comparisons of the ground truth and



(a) Ground truth (b) Recon. result (c) Close-up of (a) (d) Close-up of (b)

Fig. 12. Ground truth and reconstruction results of the synthetic data sets. (a) Ground truth of the two synthetic data sets. (b) Reconstruction results from the image stacks after adding Gaussian noise with $\sigma = 0.04477$. As can be seen from the close-up comparisons of the ground truth and reconstruction results, the reconstruction results are almost identical to the ground truth with minor differences.

reconstruction results, the reconstruction results are almost identical to the ground truth with minor differences.

Table II shows the DSC value of each image stack. The average DSC value was 0.9113, demonstrating that the segmentation results produced by the proposed method are highly overlapped with the ground truth. Table III shows the prevision-recall pair of each image stack. The averages of precision and recall were 0.9947 and 0.8964, respectively, which yielded an average F-measure of 0.9430. As mentioned above that values of DSC, precision, recall, and F-measure above 0.87 are considered good, the proposed method yields values higher than 0.89 using the evaluation

Table II. Quantitative Dice similarity coefficient measure of the synthetic data with different levels of Gaussian noise. The proposed method yields a Dice similarity coefficient average of 0.9113, demonstrating the effectiveness of the method.

	noise $\sigma = 0.0447$	noise $\sigma = 0.0632$	noise $\sigma = 0.0894$
data set 1	0.8948	0.8903	0.8868
data set 2	0.9192	0.9112	0.9040

Table III. Quantitative precision-recall measure of the synthetic data with different levels of Gaussian noise. The numbers of a pair in a field represent the values of precision and recall. The averages of precision and recall are 0.9947 and 0.8964, respectively, demonstrating the effectiveness of the method.

	noise $\sigma = 0.0447$	noise $\sigma = 0.0632$	noise $\sigma = 0.0894$
data set 1	$(0.9971, 0.8370)$	$(0.9976, 0.8121)$	$(0.9974, 0.8120)$
data set 2	$(0.9980, 0.9864)$	$(0.9955, 0.9797)$	$(0.9858, 0.9674)$

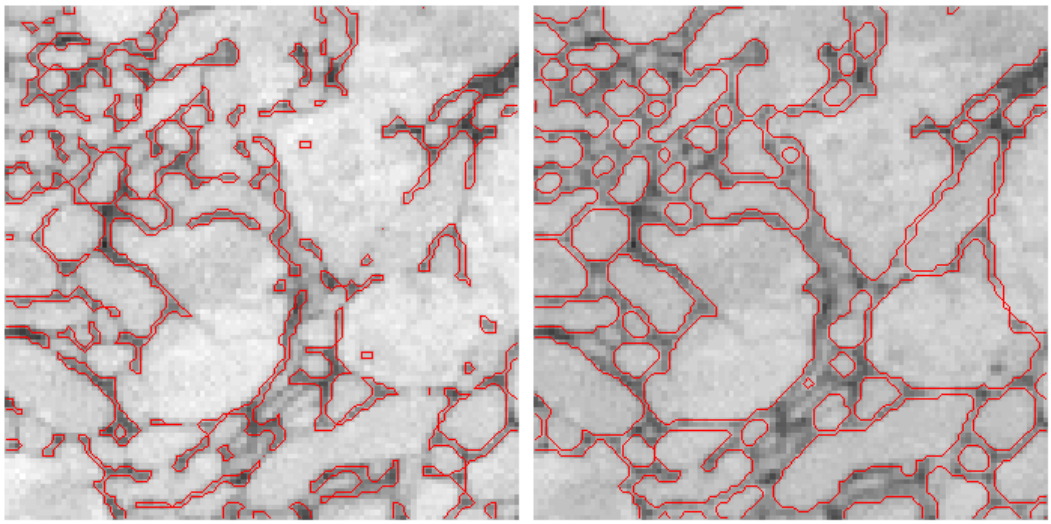
metrics, thus demonstrating the effectiveness of the method.

c. EM Data

Two stacks of serial EM images were used to evaluate the performance of the segmentation algorithm: SBFSEM and ssTEM images.

The SBFSEM Image Stack

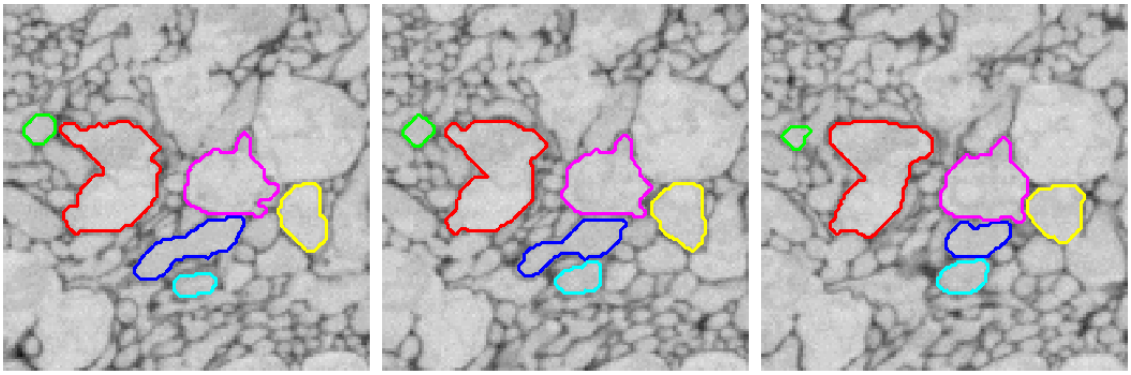
Experiments on the SBFSEM data were conducted on one image stack ($631 \times 539 \times 561$), on different parts (sub-volumes) of it. The tissue is $18.9 \times 16.1 \times 16.8$ microns in volume, with a resolution of $30 \times 30 \times 30$ nm/pixel.



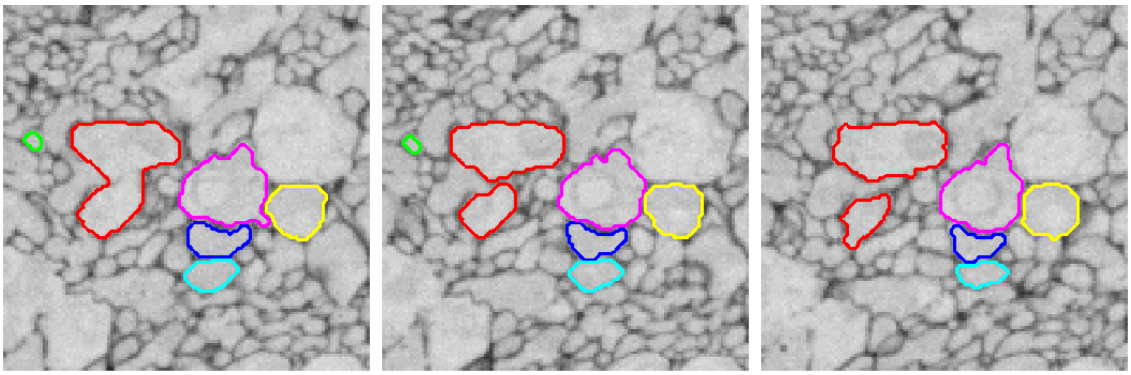
(a) Result by Kaynig et al. [34] (b) Result by the proposed method

Fig. 13. Comparison of the results obtained by Kaynig et al. [34] and by the proposed method on an SBFSEM image. Using information solely from single image causes incorrect merger of cells at locations where edge evidence cannot provide sufficient information to distinguish two cells, as shown in (a). Utilizing the geometrical information (i.e. shape prior) from the adjacent image, the proposed method can correctly delineate the boundaries of cells when the problem of boundary ambiguity occurs, as shown in (b).

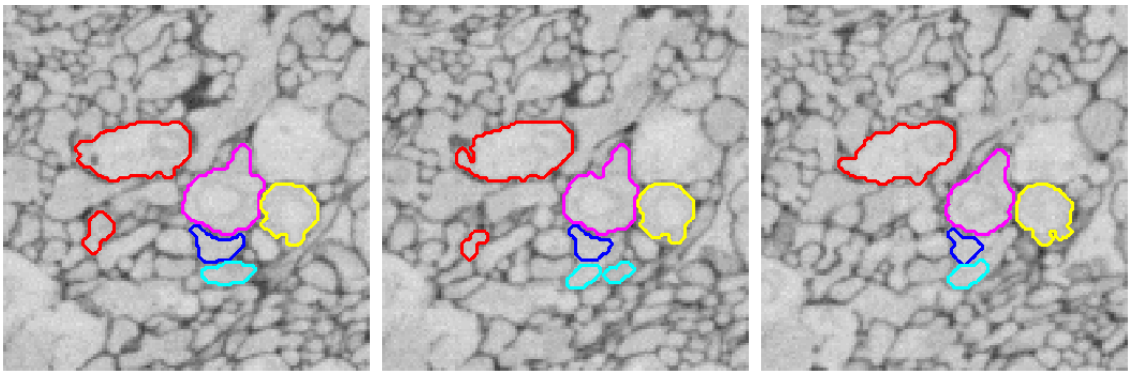
To demonstrate that the incorporation of shape information greatly improves segmentation results, the result of the proposed method is qualitatively compared to that of Kaynig et al. [34] that performs 2D segmentation using information solely from the single image to be segmented. With no additional information from neighboring images in the image stack, their method merges adjacent cells that have weak boundaries, as shown in Figure 13(a). By contrast, using the shape information from adjacent images, the proposed method can determine the correct boundaries of cells even when the problem of boundary ambiguity occurs. The result of the proposed method is shown Figure 13(b), in which adjacent cells that do not provide suffi-



(a) Slice 1 (b) Slice 3 (c) Slice 5

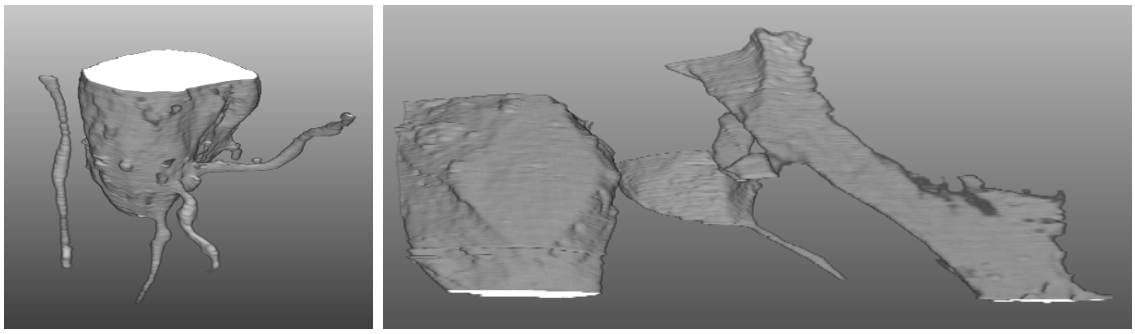


(d) Slice 7 (e) Slice 9 (f) Slice 11



(g) Slice 13 (h) Slice 15 (i) Slice 17

Fig. 14. Tracked and segmented 2D contours on selected image slices. The user manually delineates the boundaries of the regions in the first image, and the algorithm tracks and segments the selected regions in the subsequent images.



(a) (b)

Fig. 15. Reconstruction results of SBFSEM images. (a) and (b) are the partly reconstructed 3D structures, and part of a neuron can be seen in (a).

cient boundary information for determining the neuronal membranes are correctly segmented.

Figure 14 shows tracked and segmented 2D contours on selected image slices, where the user manually delineates region boundaries on the first image, and the algorithm automatically tracks and segments these 2D contours in the subsequent images. Figure 15 shows the reconstruction results, in which part of a neuron can be seen.

To quantitatively measure the performance of the proposed method, a few neurons were manually segmented using TrakEM2 [103], a tool for image stitching, registration, editing, and annotation. These manual segmentations served as the ground truth in the comparison. The average DSC value of the reconstruction results was 0.8892. The average precision and recall values of the reconstruction results were 0.9901 and 0.8202, respectively, and thus the average of F-measure was 0.8972.

The ssTEM Image Stack

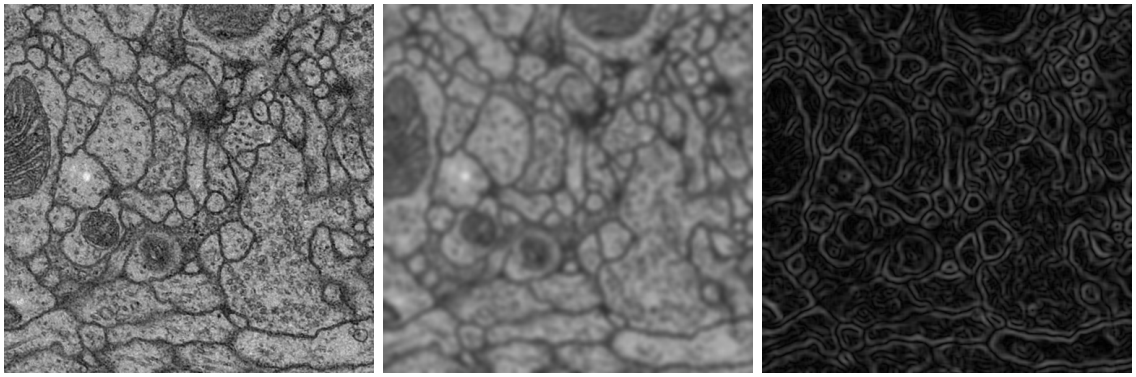
A serial section Transmission Electron Microscopy (ssTEM) data set of the Drosophila first instar larva ventral nerve cord (VNC) [24, 25] was also used for the evaluation of the proposed method. The data set contains 30 sections, each of which having a size of 512×512 pixels. The tissue is $2 \times 2 \times 1.5$ microns in volume, with a resolution of $4 \times 4 \times 50$ nm/voxel. This data set was manually delineated by an expert, and the manual segmentations served as the ground truth to which the results are compared.

Due to the imperfect image acquisition process, the raw ssTEM images exhibit inhomogeneous intensity and noise. Hence, prior to the reconstruction task being performed, a classifier was trained to enhance the neuronal membrane and remove noise in the raw ssTEM images. For training the classifier, a total of 8 features, 6 of which were adapted from those used in [103], and 2 of which were newly added, were used. The features are:

- **Pixel intensity**. The EM imaging technique stains extracellular space and suppresses intracellular organelles so the gray-scale intensity at each pixel in the raw images explicitly gives information about how to separate neurons from each other. Neuronal membranes, in general, have lower intensity values. A raw ssTEM image is shown in Figure 16(a), and the intensity ranges from 0 to 255.

- **Gaussian blur**. The smoothed image I_σ is obtained by the convolution of the raw image I with a Gaussian kernel G_σ of a certain standard deviation σ, that is,

$$I_\sigma = I * G_\sigma \ , \tag{3.19}$$



(a) ssTEM image (b) Gaussian blur (c) Sobel filter

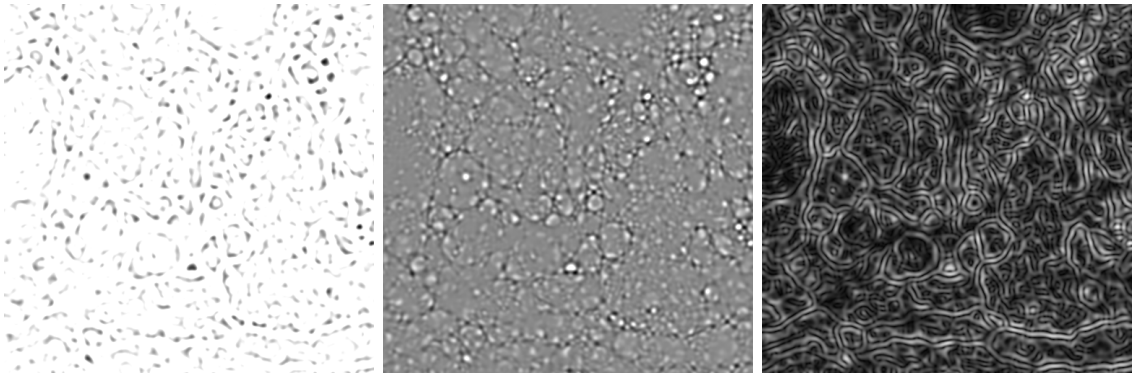
Fig. 16. Intensity, Gaussian blur, and Sobel filter. (a) A raw ssTEM image. (b) The Gaussian smoothed image obtained by the convolution of the raw image with a Gaussian kernel of a standard deviation $\sigma = 4$. (c) The magnitude of the gradients obtained by applying Sobel operators to the Gaussian smoothed image (b).

where $*$ is the convolution operator. The Gaussian kernel is defined as

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) . \tag{3.20}$$

Each raw image is convoluted with 3 Gaussian kernels, in which the standard deviation σ, respectively, equals to 1, 2, and 4. Figure 16(b) shows the Gaussian smoothed image obtained by the convolution of the raw image with a Gaussian kernel of σ set to 4.

- **Sobel filter**. The smoothed image I_σ is then convoluted with two 3×3 Sobel operators to calculate the derivatives in the horizontal and vertical directions. The magnitude of the gradients is used as a feature, as shown in Figure 16(c).



(a) Trace (b) Determinant (c) Value

Fig. 17. Trace, determinant, and value of Hessian matrix of an ssTEM image.

- **Hessian-based features**. The Hessian matrix at a given point p is defined as

$$H(p) = \begin{bmatrix} L_{xx}(p) & L_{xy}(p) \\ L_{xy}(p) & L_{yy}(p) \end{bmatrix}, \tag{3.21}$$

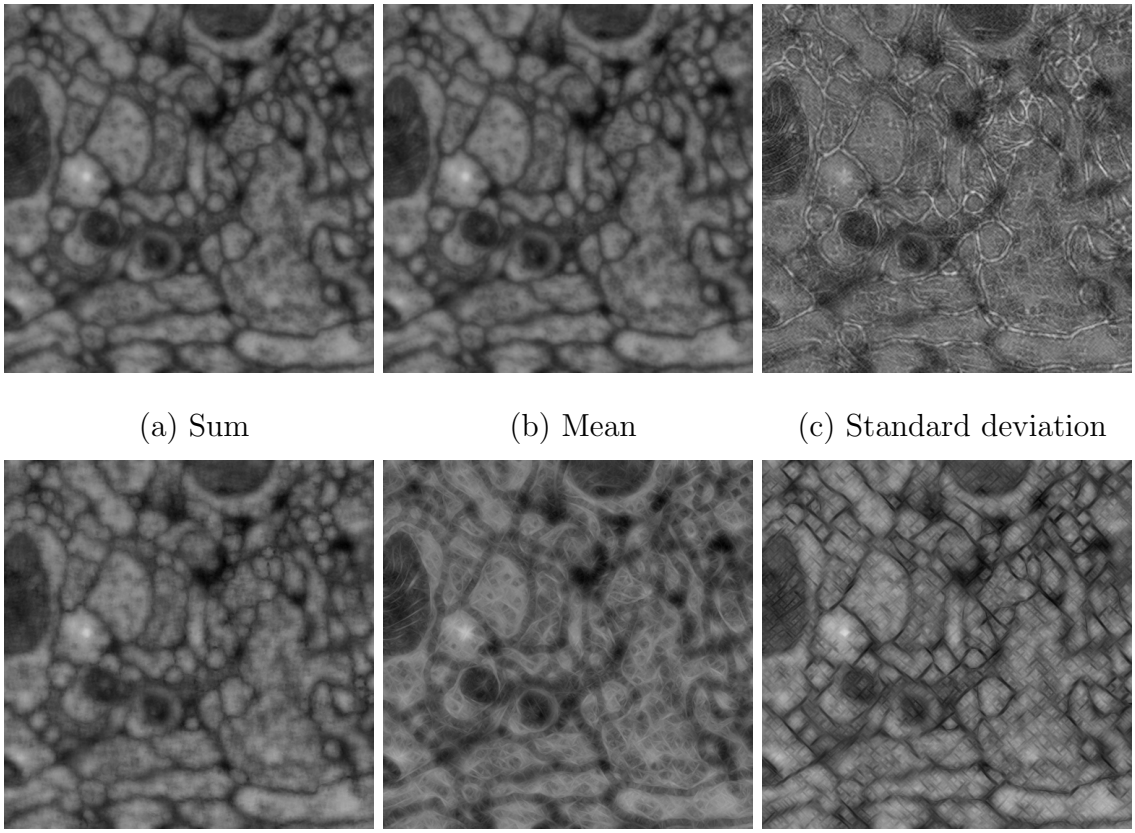
where L_{xx} is the second partial derivative in the x direction, L_{xy} is the mixed partial second derivative in the x and y directions, and $L(p) = I(p) * G_\sigma$. Thus, the trace TR (sum of the diagonal elements) and determinant DET of the Hessian matrix are given as

$$TR = L_{xx} + L_{yy} \tag{3.22}$$

and

$$DET = L_{xx}L_{yy} - L_{xy}^2 . \tag{3.23}$$

These two features are invariant to the selection of x and y. Besides, another value, computed as $\sqrt{L_{xx}^2 + L_{xy}L_{yx} + L_{yy}^2}$, is also extracted for training. Shown in Figures 17(a), 17(b), and 17(c), are the trace, determinant, and value, respectively.

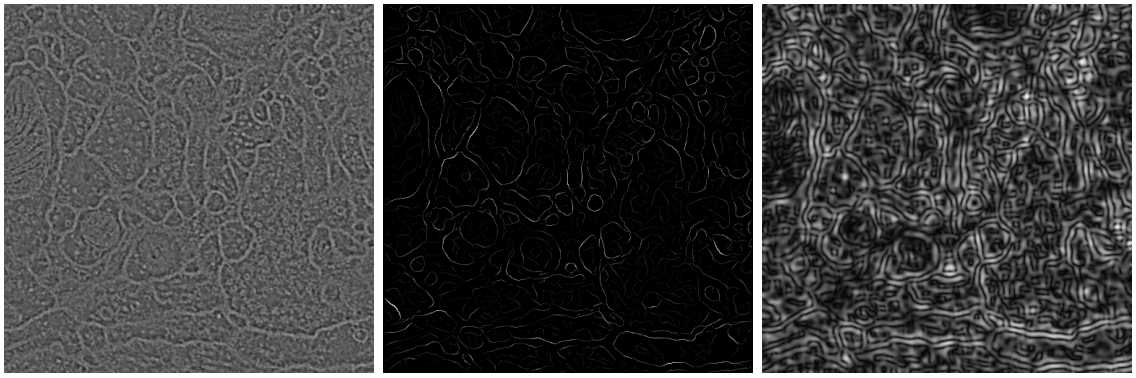


(a) Sum (b) Mean (c) Standard deviation

(d) Median (e) Maximum (f) Minimum

Fig. 18. Sum, mean, standard deviation, median, maximum, and minimum of the pixels in each image obtained by z-projecting 12 filtered images.

- **Membrane projections**. A filter of a size of 19×19, whose center column is set to 1 and 0 elsewhere, is designed to detect the neuronal membranes. In order to detect membranes in various directions, this filter is rotated clockwise every 15 degrees, resulting in a set of 12 different filters. The raw image is then respectively convoluted with these 12 created filters. By doing so, it generates 12 filtered images that indicate the possible locations of neuronal membranes. These 12 images are then *z-projected*, and the following features are extracted:



(a) DoG (b) PB (c) Curvedness

Fig. 19. Difference of Gaussian (DoG), probability of boundary (PB), and curvedness of an image.

- sum of the pixels in each image as shown in Figure 18(a).
- mean of the pixels in each image as shown in Figure 18(b).
- standard deviation of the pixels in each image as shown in Figure 18(c).
- median of the pixels in each image as shown in Figure 18(d).
- maximum of the pixels in each image as shown in Figure 18(e).
- minimum of the pixels in each image as shown in Figure 18(f).

- **Difference of Gaussian (DoG)**. The DoG image I_D is given by

$$I_D = I * D, \tag{3.24}$$

where I is the raw image, and D is the DoG filter and defined as the difference of two Gaussian kernels with different standard deviations. Mathematically, the DoG filter is

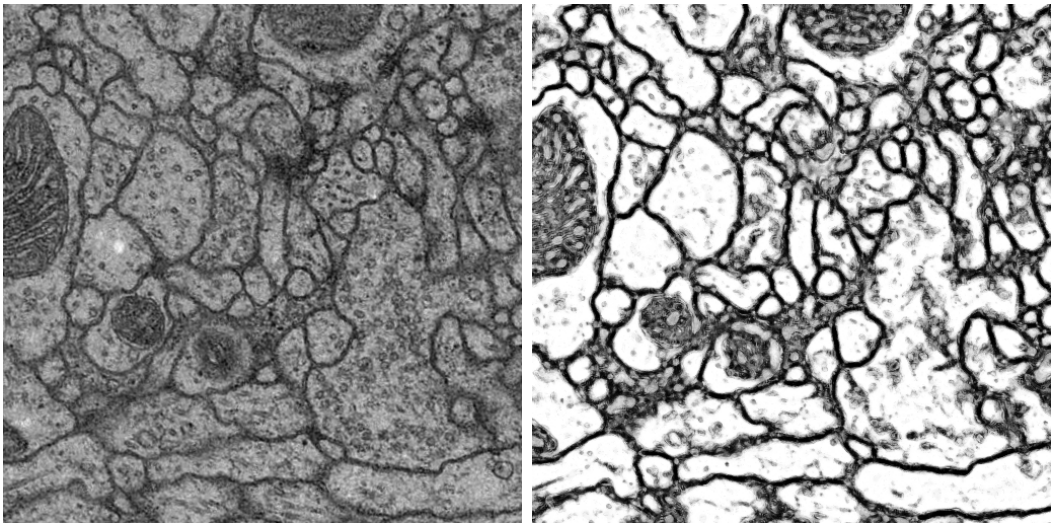
$$D(x, y) = G_{\sigma_1}(x, y) - G_{\sigma_2}(x, y) \ , \tag{3.25}$$

where G_σ is a Gaussian kernel given in Equation 3.20. Figure 19(a) shows the resulting DoG filtered image.

- **Probability of Boundary**. This is a newly added feature. The probability of a pixel belonging to a boundary is the output of a classifier that provides the posterior probability of a boundary at each pixel. The classifier is trained by combining local brightness, color, and texture features in order to accurately detect the boundary [104]. The probability of a boundary is shown in Figure 19(b).

- **Curvedness**. This is also a newly added feature. The curvedness is calculated by using Equation 3.12 for each smoothed image. The resulting curvedness of an image is shown in Figure 19(c).

For each pixel of an image, the above features are extracted. Using these features, a random forest classifier [103, 105] was trained for the detection of neuronal membranes. By definition, random forest is an ensemble of decision trees, each of which is independently trained using randomly selected samples. This ensemble of tress makes random forest retain the advantages of a decision tree, namely handling of missing values and resistance to outliers, while increasing accuracy [106]. Additionally, it has few parameters and is faster to train and to predict. When random forest is applied to a classification problem, its prediction is the class which most of individual trees agree on (i.e. the majority vote of all individual trees). Figure 20(b) shows the output of the classifier, in which the neuronal membranes are greatly enhanced, and the noise is removed compared to the raw image shown in Figure 20(a). The resulting images processed by the random forest classifier are the inputs to the proposed method.

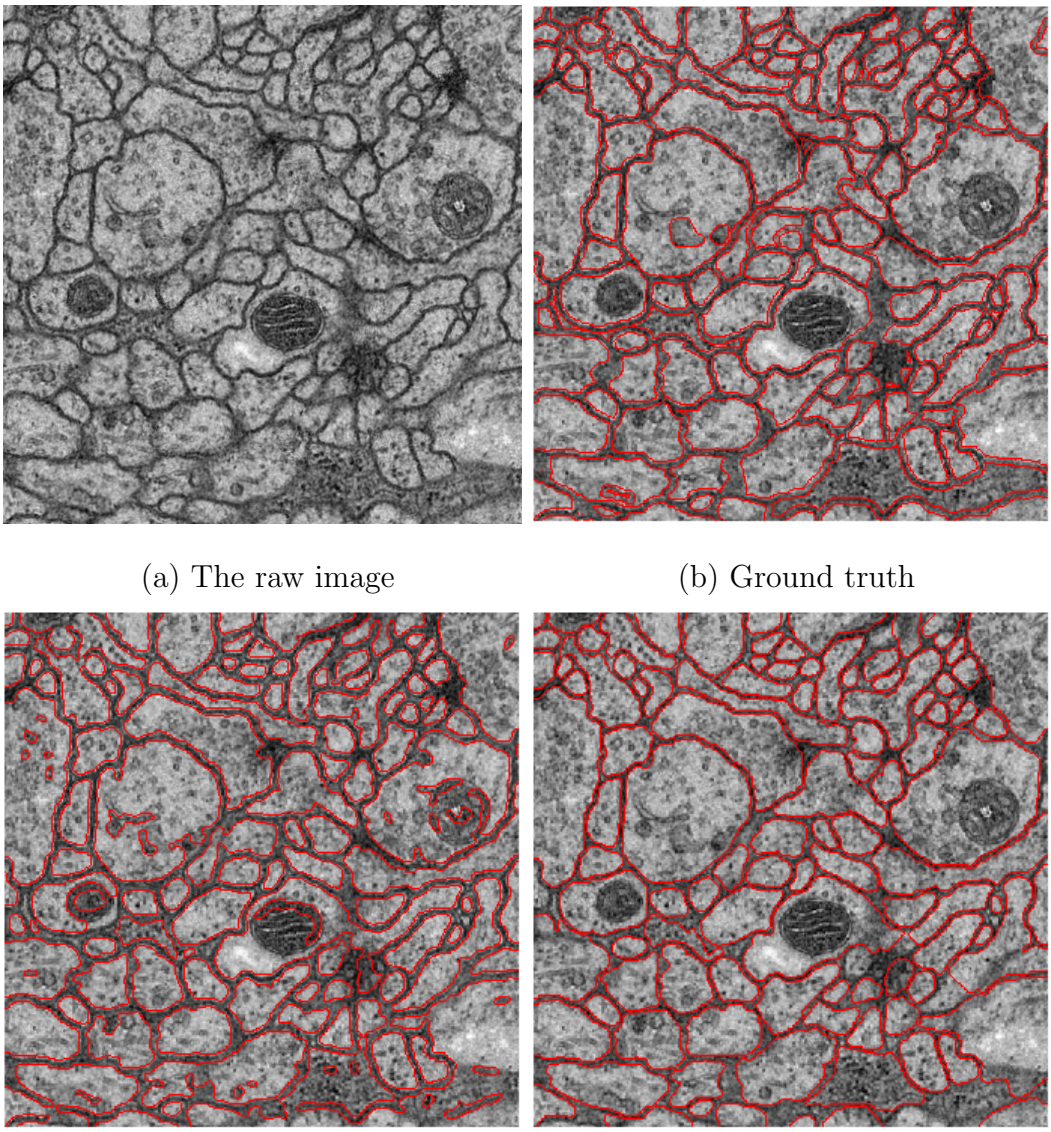


(a) A raw ssTEM image (b) The enhanced image

Fig. 20. Comparison of a raw ssTEM image and its enhanced image obtained by a random forest classifier. The random forest classifier greatly enhances the contrast between neurons' interior and the neuronal membranes.

The proposed method was compared to the method by Kaynig et al. [34]. Figure 21 shows the results by the proposed method compared to Kaynig et al. and manual segmentation. Both methods produced comparable results when qualitatively compared to the ground truth. Using DSC as the measure metric, the proposed method produced a value of 0.7966, which is slightly higher than the value of 0.7853 obtained by Kaynig et al. F-measure of the proposed method was 0.7942, also slightly better than the value of 0.7853 obtained by Kaynig et al. Table IV shows the detailed comparison between the method by Kaynig et al. and the proposed method.

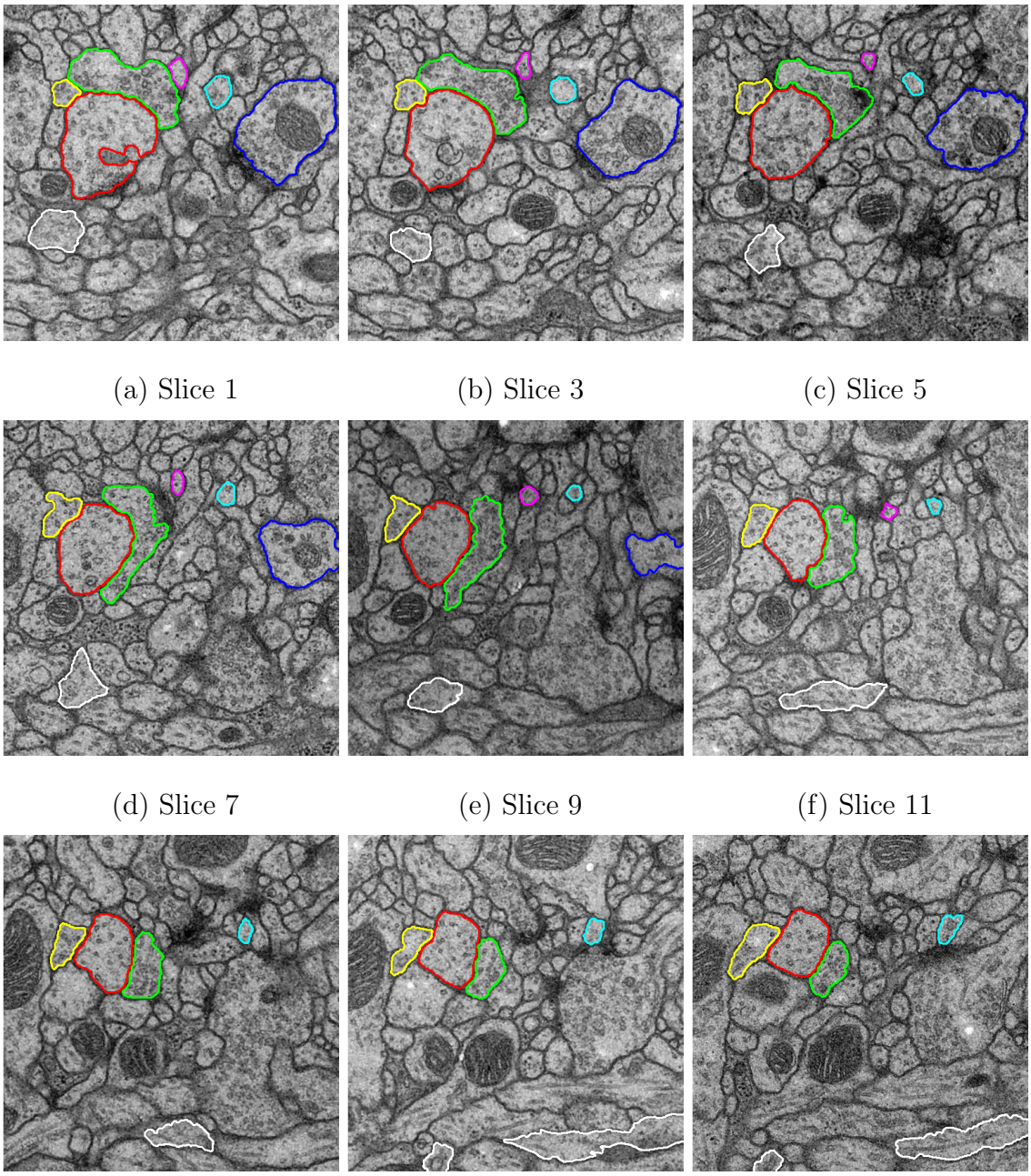
Same as the experiments on the SBFSEM images, initial segmentations provided by a user are required in order for the segmentation algorithm to start tracking and segmenting the neuronal contours. Figure 22 shows the segmentation and tracking results of selected regions on the ssTEM images, where different colors in the same



(a) The raw image (b) Ground truth

(c) Result by Kaynig et al. [34] (d) Result by the proposed method

Fig. 21. Comparison of results obtained by Kaynig et al. [34] and by the proposed method and the ground truth on a sample ssTEM image. By qualitative comparison, the results of Kaynig et al. and the proposed method are comparable to the manual segmentation.



(a) Slice 1 (b) Slice 3 (c) Slice 5

(d) Slice 7 (e) Slice 9 (f) Slice 11

(g) Slice 13 (h) Slice 15 (i) Slice 17

Fig. 22. Tracking results of selected regions on ssTEM images. Different colors in the same image indicate different neurons, and same color in different slices represent the same neuron. The proposed method successfully segments and tracks selected regions through cross-sectional images.

Table IV. Quantitative comparison of the method by Kaynig et al. [34] and the proposed method. The proposed method yielded a DSC value of 0.7966 and F-measure of 0.7942, both of which are slightly higher than the DSC value of 0.7853 and F-measure of 0.7853 obtained by using the method of Kaynig et al.

	DSC	Precision	Recall	F-measure
Kaynig et al. [34]	0.7853	0.7310	0.8484	0.7853
Proposed method	0.7966	0.8127	0.7766	0.7942

image indicate different neurons, and the same color in different slices represents the same neuron. Figure 23 shows a few reconstruction results. The ssTEM image stack contains only 30 slices so the reconstructions do not show whole neuron structures. However, as can been seen, the proposed method successfully segments and correctly tracks regions through cross-sectional images.

B. 3D Segmentation with Estimated Shape Prior

This section describes a 3D segmentation method that extends the aforementioned 2D segmentation into a 3D. The 3D segmentation method incorporates an estimated 3D shape prior into the energy function, which serves as a constraint in the segmentation process. The proposed method consists of two main parts. First, different from other methods that derive the shape prior in an offline phase, the shape prior of the objects is estimated directly by extracting medial surfaces from the data set. Second, the 3D image segmentation problem is posed as Maximum A Posteriori (MAP) estimation of Markov Random Field (MRF). The energy function to be minimized contains three constituents: the estimated shape prior, the flux of the image gradients, and the gray-scale intensity.

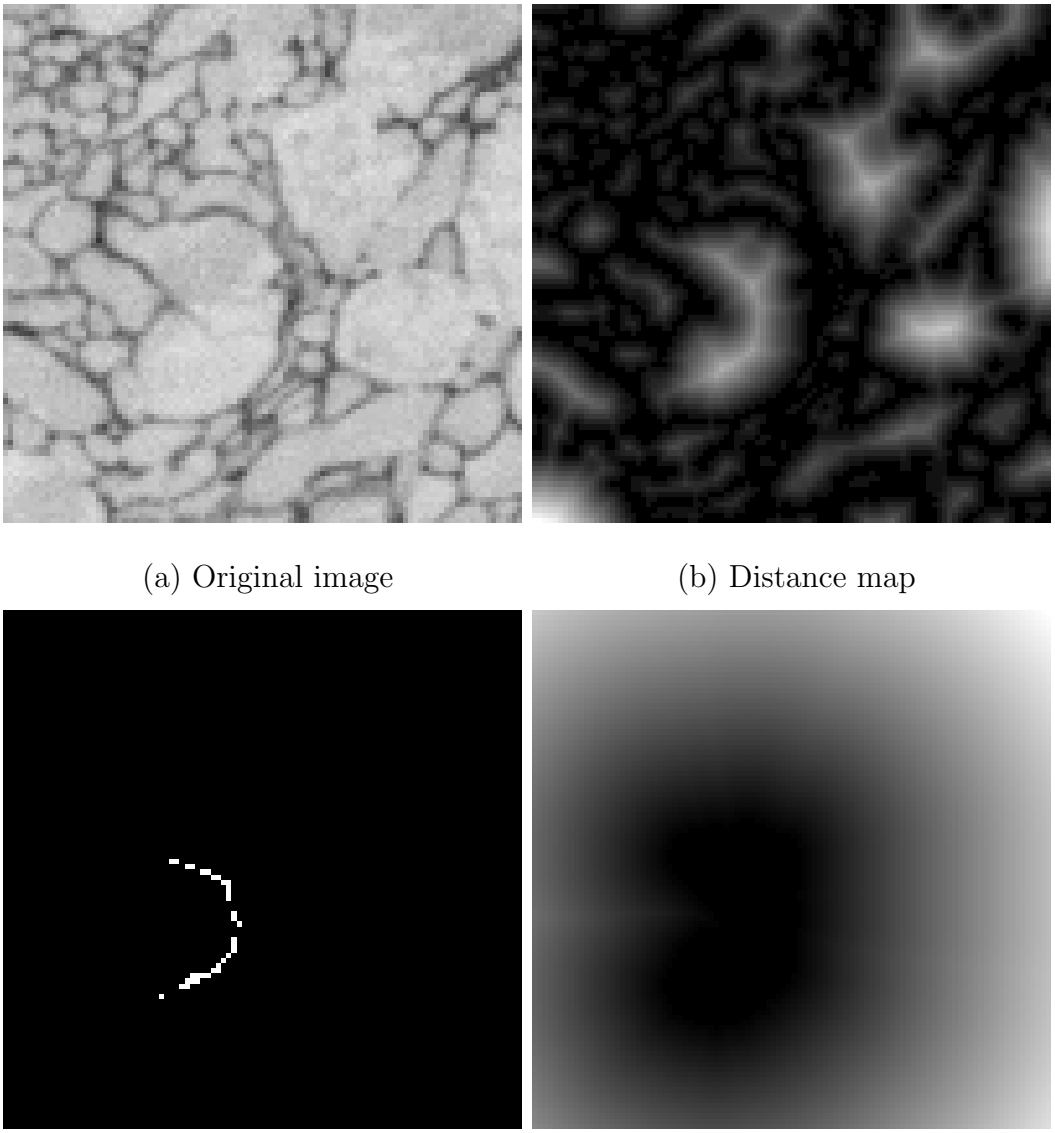
(a) (b)

Fig. 23. Reconstruction results of the ssTEM data. Because the ssTEM image stack contains only 30 slices, the reconstruction results do not show a whole neuron structure.

1. Symmetric Shape Prior Estimation

Shape symmetry is an important visual feature in image understanding, and the use of symmetry, which provides shape information of an object, can significantly improve the segmentation [107, 108]. Because the anatomical structures, such as axons, dendrites, and soma, exhibit locally symmetric shapes to the medial axis that is also referred to as the skeleton and is commonly used for shape representation, one way to estimate the shape prior is to utilize this intrinsic property of anatomical structures. Following the definition introduction by Blum [109], the medial axis of an object is a set of points that is equidistant to its boundaries. The medial axis of a 3D object is generally referred to as the medial surface.

Approaches to extracting medial surface include distance field based methods [110], topological thinning, gradient vector flow methods [111], and others [112]. Here, the method proposed by Bouix et al. [110] is used to extract the medial surface.



(a) Original image (b) Distance map

(c) Skeleton (d) Shape prior

Fig. 24. Method of shape prior estimation. (a) An SBFSEM image. (b) The distance map computed from the binary image. (c) The extracted skeleton (white curves) from the distance map. (d) The estimated shape prior. Dark is the expanded region, and bright indicates the points outside the expanded region which are represented by a distance function.

Gray-scale images are first converted to binary images, and a Euclidean distance function to the nearest boundary is computed at each voxel, as shown in Figure 24(b). Guiding the thinning procedure by exploiting properties of the average outward flux of the gradient vector field of a distance transform, the resulting medial surface for a particular object is shown in Figure 24(c). Finally, the estimated shape is obtained by first expanding each point in the extracted medial surface with the shortest distance to the boundary. The points outside the expanded region are represented by a distance function

$$D(p) = \|p - s_p\| , \tag{3.26}$$

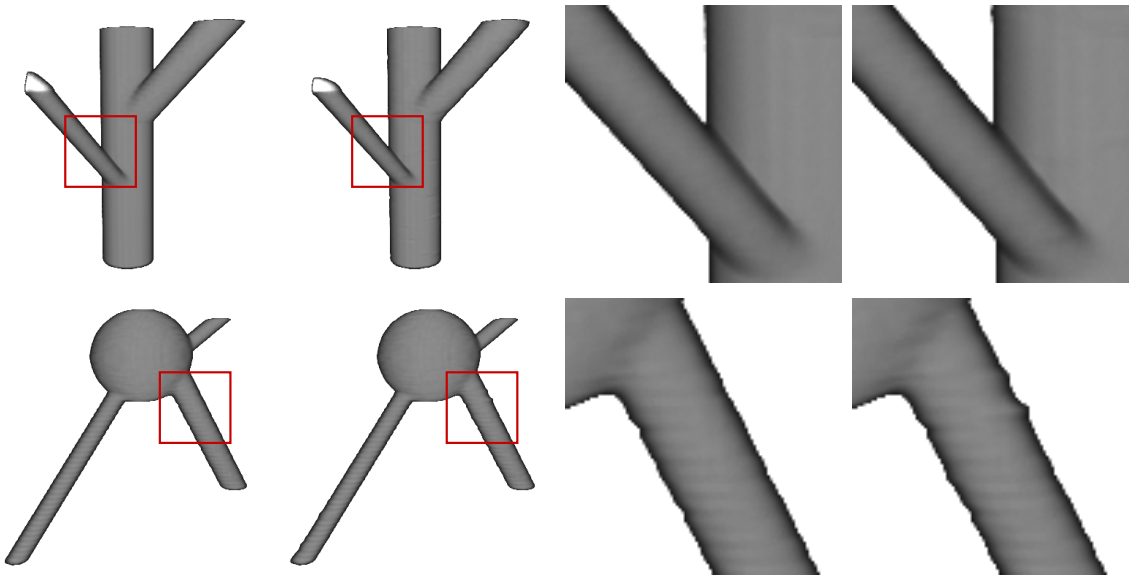
where $\|p - s_p\|$ represents the Euclidean distance from point p to the nearest point s_p in the expanded region. Shown in Figure 24(d) is the estimated shape prior for the object in Figure 24(c).

2. Unary Potential: Flux and Estimated Shape Prior

Similar to the aforementioned 2D approach, the unary potential includes the flux and estimated shape prior shown in Figure 24(d). The way to integrate these two constituents is the same as that in the 2D case (i.e. Equation 3.7). The estimated shape prior acts as a geometrical constraint in the minimization process.

3. Piecewise Potential: Image Intensity

The piecewise potential captures the contextual information between a point and its neighboring points. To construct a graph in 3D, a point not only connects to its neighbors in the same plane (x-y plane), but also connects to the points in the adjacent images (z direction). Generally, the 6-, 18-, or 26-neighborhood system is commonly used in 3D. Here, the 6-neighborhood system was used in the experiments.



(a) Ground truth (b) Recon. result (c) Close-up of (a) (d) Close-up of (b)

Fig. 25. Ground truth and reconstruction results of the synthetic data sets. (a) Ground truth of the two synthetic data sets. (b) Reconstruction results from the image stacks in which Gaussian noise with $\sigma = 0.04477$ was added. As can be seen from the close-up comparisons of the ground truth and the reconstruction results, the reconstruction results are almost identical to the ground truth with minor differences.

4. Experimental Results

Similar to the experiments carried out for the aforementioned tracking and segmentation framework, the same synthetic data and SBFSEM data were used in the experiments. Their results are shown in the subsequent sections.

a. Synthetic Data

The reconstruction results of the two synthetic image stacks are shown in Figure 25(b), and their ground truth is given in Figure 25(a) accordingly. As can been seen from the close-up comparisons of the reconstruction results and the ground truth, the

Table V. Quantitative Dice similarity coefficient measure of the synthetic data with different levels of Gaussian noise. The proposed method yields a DSC average of 0.9748, demonstrating the effectiveness of the method.

	noise $\sigma = 0.0447$	noise $\sigma = 0.0632$	noise $\sigma = 0.0894$
data set 1	0.9917	0.9671	0.9397
data set 2	0.9962	0.9875	0.9664

Table VI. Quantitative precision-recall measure of the synthetic data with different levels of Gaussian noise. The numbers of a pair in a field represent the values of precision and recall. The averages of precision and recall are 0.9659 and 0.9853, respectively, demonstrating the effectiveness of the method.

	noise $\sigma = 0.0447$	noise $\sigma = 0.0632$	noise $\sigma = 0.0894$
data set 1	$(0.9857, 0.9982)$	$(0.9494, 0.9870)$	$(0.9103, 0.9752)$
data set 2	$(0.9987, 0.9937)$	$(0.9947, 0.9806)$	$(0.9565, 0.9770)$

reconstruction results are almost identical to the ground truth with minor differences.

Table V shows the DSC values for each image stack. The average DSC was 0.9748. Table VI shows the precision and recall pair, the mean values of the precision and recall were 0.9656 and 0.9853, respectively. As a result, F-measure was 0.9755. The 3D method yielded values of DSC, precision, recall, and F-measure above 0.9, which is considered good as compared to the baseline value of 0.87.

The mean computation time using a Matlab implementation of the proposed approach to process a synthetic image stack ($100 \times 100 \times 100$) on a standard PC with Core 2 Duo CPU 2.2 GHz and 2 GB memory was 20 seconds.

(a) (b) (c)

Fig. 26. Reconstruction results of the proposed method. (a) and (b) show parts of neurons, and (c) shows the elongated axon structures.

b. SBFSEM Data

The reconstruction results of the proposed method are shown in Figure 26, where Figure 26(a) and Figure 26(b) show parts of neurons, and Figure 26(c) shows the elongated structures.

To quantitatively measure the performance of the proposed method, a few neurons were manually segmented using TrakEM2 [103]. The manual segmentations served as the ground truth in the comparison. Again, F-measure and DSC were used as the evaluation metrics. The average precision and recall values of the reconstruction results shown in Figure 26 were 0.9660 and 0.8424, respectively, and thus the average of F-measure was 0.9. The average DSC value of the reconstruction results was 0.8918, showing that the proposed method can reconstruct the neuronal structures from the SBFSEM images.

C. Summary

This chapter presented new segmentation methods aiming to reconstruct neuronal processes from stacks of serial EM images. The geometrical information of adjacent images (i.e. shape priors) greatly improves the segmentation accuracy. With the help of the shape priors, together with the flux of image gradients and image gray-scale intensity, the proposed segmentation approaches effectively reconstruct the anatomical structures from densely packed EM images.

CHAPTER IV

INTERACTIVE EDITING THROUGH MULTIPLE CHOICE AND GRAPH CUTS

A great majority of existing segmentation algorithms are designed for the automation of the segmentation pipeline. However, fully automated segmentation algorithms sometimes yield incorrect results even when their parameters are optimally tuned, which is mainly because of their failing to capture all variations posed by the data sets to be processed. Thus, erroneous segmentations would inevitably arise, and they may require manual correction.

This chapter presents interactive editing tools to refine segmentation results with minimal interaction. The framework includes two editing methods: (1) editing through multiple choice that provides a set of segmentation alternatives from which a user can select an acceptable one, and (2) interactive editing through graph cuts that allows a user to edit segmentation results by interactively placing editing strokes. These tools facilitate the user to correct segmentations easily and quickly.

A. Editing through Multiple Choice

Image segmentation is an ill-posed problem. A particular feature used or parameter choice for a segmentation algorithm strongly affects the quality of segmentations [113], thus many researchers have considered multiple segmentations for an image for the segmentation task [114, 115]. A general way to obtain multiple segmentations of an image is to execute a segmentation algorithm several times by using different parameter settings. Rather than simply tuning parameters of a segmentation method, a more sophisticated approach that integrates multiple cues is usually used to provide multiple segmentations. In this approach, each cue is integrated for the

generation of a confidence map that indicates the probability of each pixel belonging to a specific label. By thresholding the confidence map using different values, a pool of segmentation alternatives are generated. The user can choose the most acceptable one, if available, among the automatically generated segmentations before starting any manual editing. Allowing the user to select the desired segmentation prior to manual editing will minimize user interaction and simplify the editing process. This approach is detailed in the subsequent sections.

1. Cues for Confidence Map Generation

It has been shown in [116] that the visual system integrates different attributes (e.g. luminance, color, motion, and texture) for contour localization because all attributes play an essential role for a contour localization task. This motivates the use of multiple cues, analogous to different attributes in localization of contours, to generate a confidence map that indicates the degree of a pixel belonging to a particular label. These cues are:

- **Probability of boundary (PB)**. Boundary is an important cue for distinguishing different segments in the image segmentation task. This cue takes the output of a classifier that provides the posterior probability of a boundary at each pixel. The classifier is trained by combining local brightness, color, and texture features in order to accurately detect the boundary [104]. Figure 29(a) shows the probability of a boundary of an image.

- **Random walker segmentation (RW)**. In contrast to a hard segmentation, that is, a pixel belonging to either the foreground (1) or not (0) for the binary case, random walker segmentation [64] produces a soft segmentation. Figure 27 illustrates how a segmentation is generated by the random walker segmentation

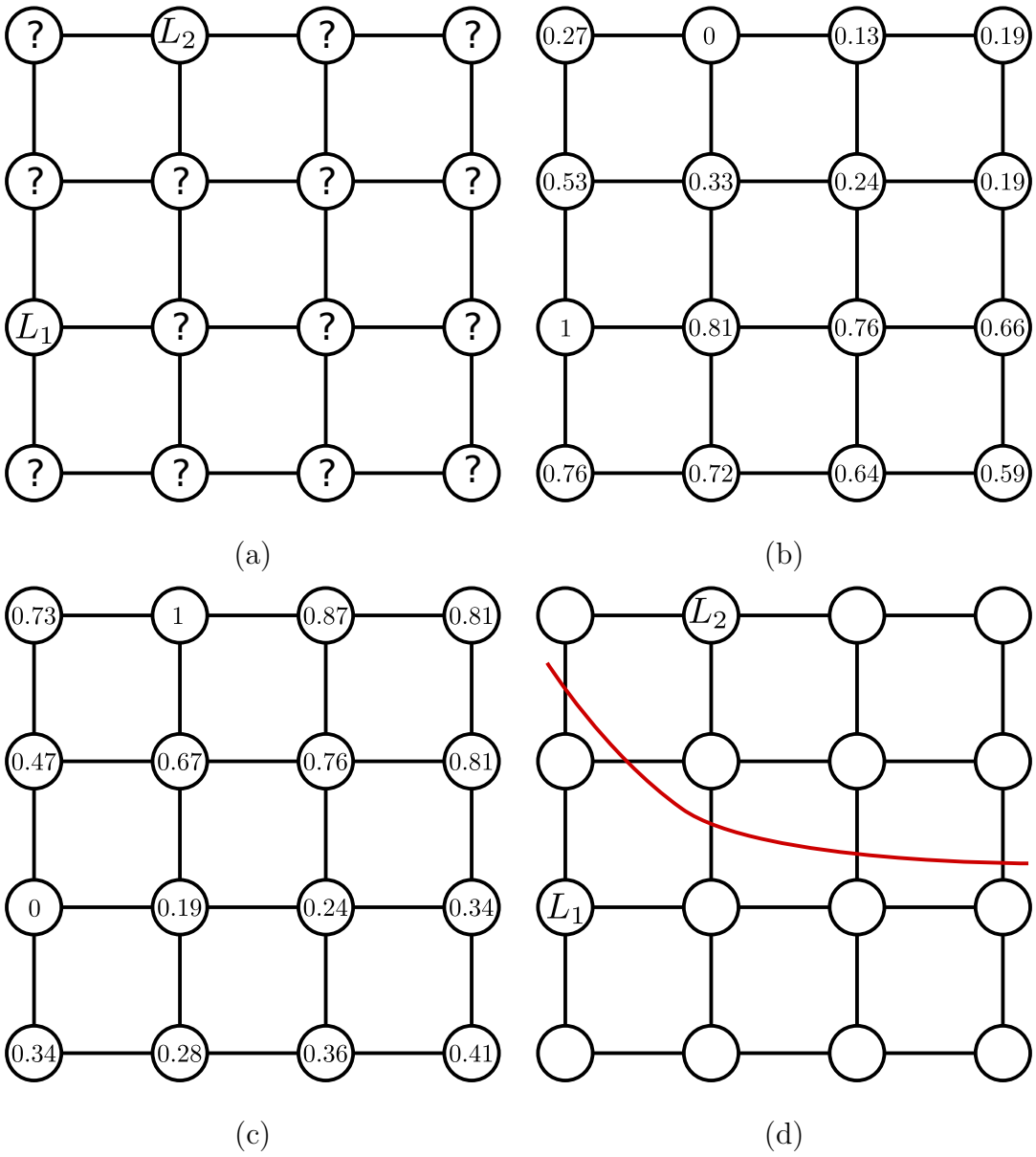
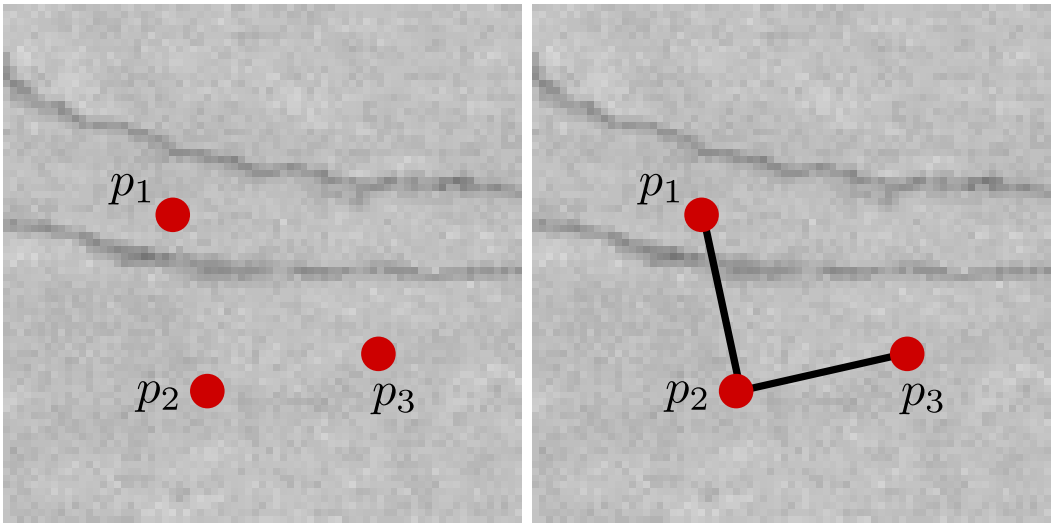


Fig. 27. Illustration of random walker segmentation. (a) An image with two different labels, L_1 and L_2. (b) Probability that a random walker starting from each unlabeld node first reaches to the node with label L_1. (c) Probability that a random walker starting from each unlabeld node first reaches to the node with label L_2. (d) The segmentation obtained by assigning each point to the label that is of the largest probability.

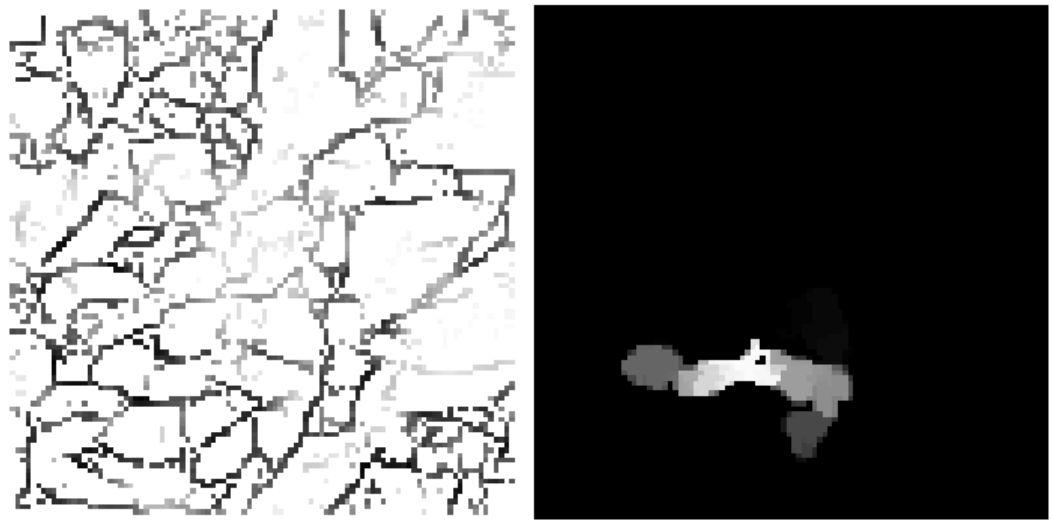


(a) (b)

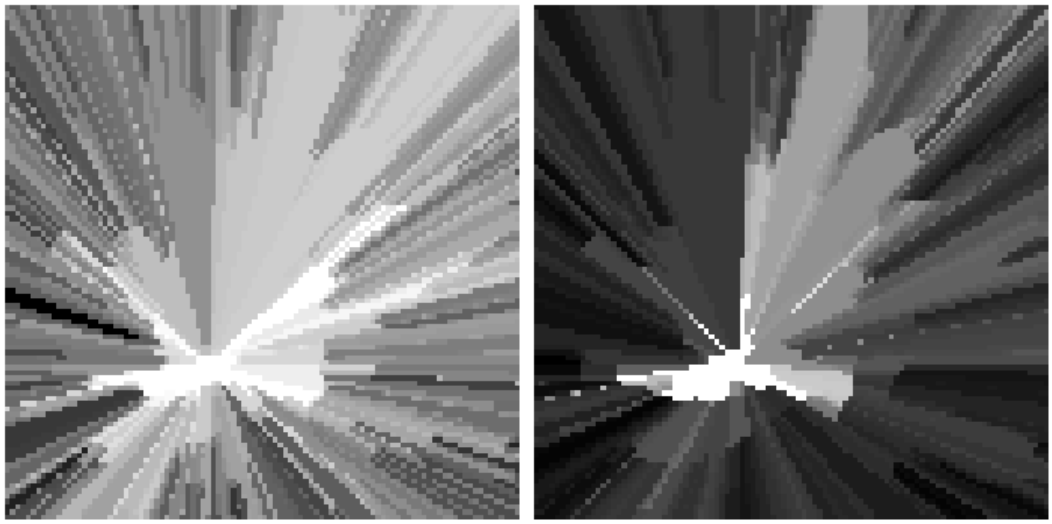
Fig. 28. Illustration of how the affinities between points are calculated using intervening contours. (a) An image with 3 points, p_1, p_2, and p_3. (b) The affinity between two points is measured based on the maximum gradient magnitude (or other measurements) of a straight-line path between them. For example, because the straight-line path between points p_1 and p_2 crosses the boundary, the affinity between them is lower than that between points p_2 and p_3, which lie on the same side of the boundary. This indicates that points on the two different sides of a boundary are more likely to belong to different segments.

algorithm. Initially, two different labels, L_1 and L_2, are specified by the user as shown in Figure 27(a). The probabilities that a random walker starting from each unlabeld node first reaches to the labeled nodes L_1 and L_2 are shown in Figure 27(b) and Figure 27(c), respectively. The segmentation is obtained by assigning each point to the label that is of the largest probability, which is shown in Figure 27(d). Figure 29(b) shows the result of applying the random walker segmentation to an SBFSEM image.

- **Intervening contours (IC)**. The *intervening contour* [117] concept suggests



(a) PB (b) RW



(c) IC(1) (d) IC(2)

Fig. 29. Multiple cues used for generation of a confidence map. (a) Probability of a boundary. (c) The result of the random walker algorithm. (c) The affinity obtained by using gradient magnitude as a measurement in the intervening contour cue. (d) The affinity obtained by using the probability of a boundary as a measurement in the intervening contour cue.

that pixels on the two different sides of a boundary are more likely to belong to different segments. Given two pixels on an image, the affinity between them is measured as the maximum gradient magnitude (or other measurements) of a straight-line path between them. Figure 28 illustrates how affinities are computed by using intervening contours. Because the straight-line path between points p_1 and p_2 crosses the boundary, their affinity is much lower than that between points p_2 and p_3, which lie on the same side of the boundary. Figure 29(c) (IC1) shows the affinities between a point inside an object and any other point, which are computed based on the maximum gradient magnitude. In addition to using the gradient magnitude as a measurement, the probability of a boundary is also used to compute the affinities between pixels. Figure 29(d) (IC2) shows the result of considering the probability of a boundary as a measurement in computing an intervening contour cue.

2. Cue Combination

Before all of the individual cues are combined, their values are normalized to $[0, 1]$. Let $P\left(c_p^k \middle| I\right)$ be the value at pixel p produced by cue k. The confidence value at pixel p is defined as a linear combination of all cues, which is

$$P\left(c_p|I\right) = \sum_k w_k P\left(c_p^k|I\right) \ , \tag{4.1}$$

where w_k is the relative importance of cue k, and sum of all w_k is equal to 1. Currently, the values for all w_k were set empirically. A better way to obtain the weight of each cue is to find a weight combination that yields the best results through comparing the results obtained by various weight combinations to the ground truth, which can be achieved by training a classifier. The confidence map shown in Figure 30(b) is a combination of Figure 29(a) through Figure 29(d). Note that the brighter color

indicates a higher probability of belonging to an object.

3. Segmentation Alternatives Generation

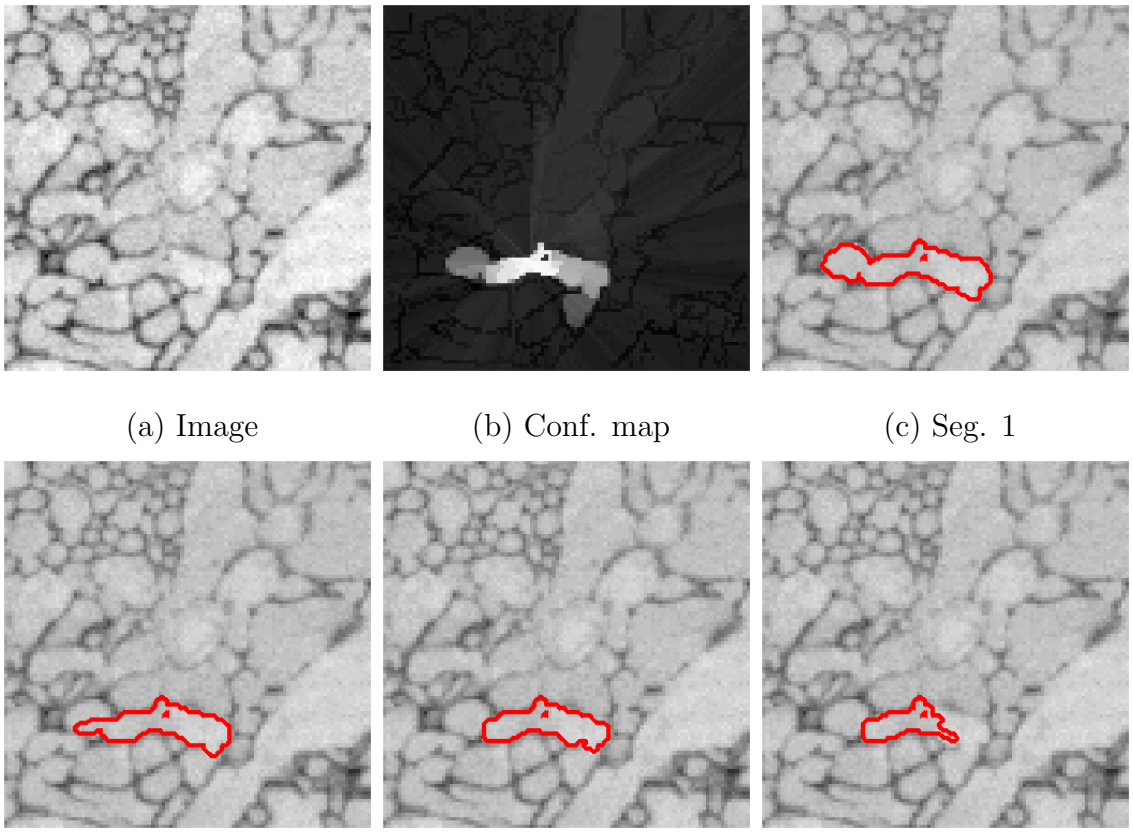
To generate multiple segmentation alternatives of an image, the confidence map is thresholded at different values. Figure 30(c) through Figure 30(f) show the segmentation alternatives obtained by using the threshold values between 0.4 and 0.7. The user can choose the most acceptable segmentation, if one existed, from these alternatives before applying any editing. This reduces the amount of time for user interaction.

4. Experimental Results

The experiments were carried out on the SBFSEM image stack. A major challenge in reconstructing neuronal morphology from SBFSEM image data lies in segmenting densely packed cells that have blurred boundaries mainly resulting from the imperfect image acquisition process. As a result, user editing is key to resolving the boundary ambiguities, in which case a segmentation algorithm fails to produce a satisfactory result.

Thirty incorrect segmentations produced by an automated segmentation algorithm were used in the experiment. A few alternative segmentations generated by using the editing through multiple choice method are shown in Figure 31. These examples demonstrate that the generated alternatives contain a few acceptable segmentations, showing that the confidence map obtained by integrating different cues is reliable.

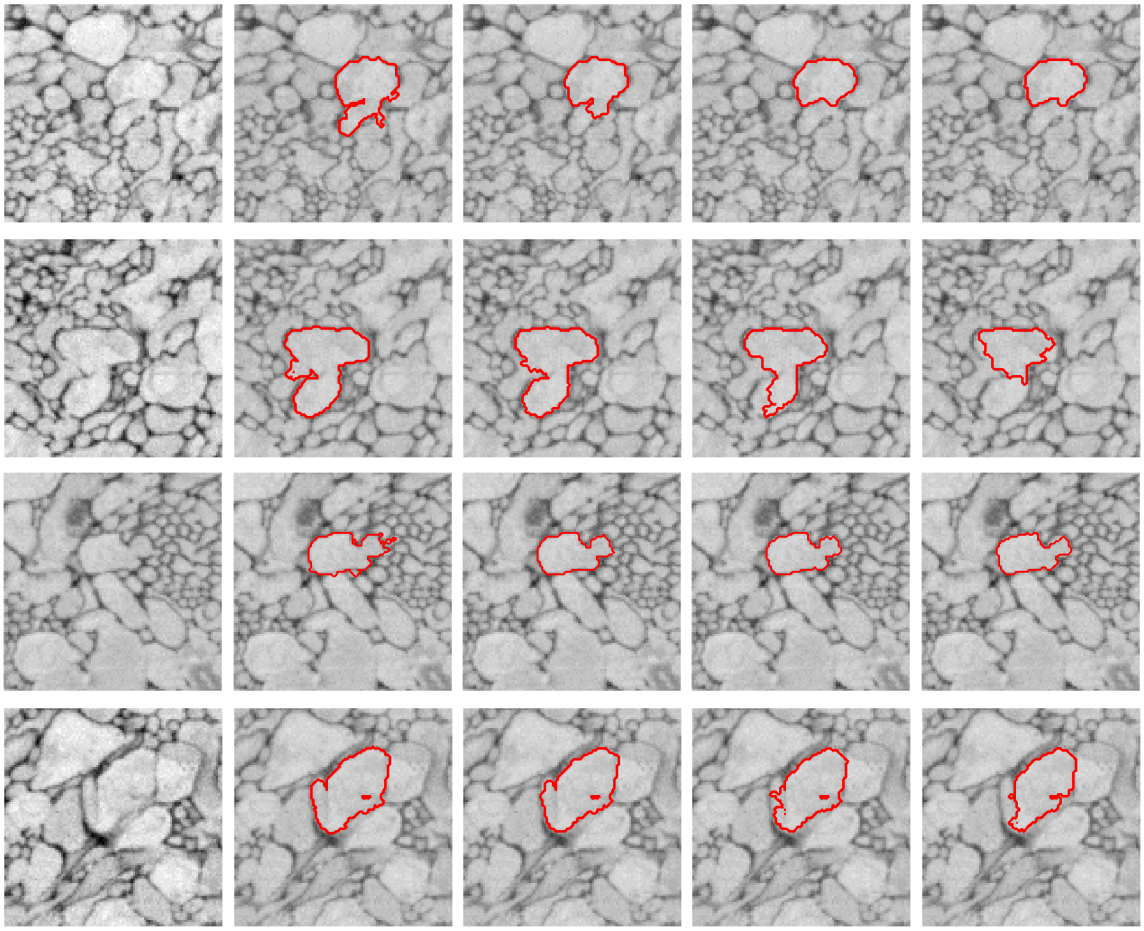
When the editing through multiple choice method was applied to refining incorrect segmentations, around 15 out of 30 segmentation errors can be corrected (i.e., the user can obtain an acceptable segmentation from the generated alternatives). For quantitative evaluation, Dice similarity coefficient (DSC) and F-measure were used to



(a) Image (b) Conf. map (c) Seg. 1

(d) Seg. 2 (e) Seg. 3 (f) Seg. 4

Fig. 30. Cue combination to generate a confidence map and multiple segmentations. (a) is the original image. (b) is the confidence map generated by a linear combination of different cues. Note that the brighter color indicates a higher probability of belonging to an object. (c), (d), (e) and (f) are segmentations obtained by thresholding the confidence map at the values of 0.4, 0.5, 0.6 and 0.7, respectively.



(a) Image (b) Seg. 1 (c) Seg. 2 (d) Seg. 3 (e) Seg. 4

Fig. 31. Examples of generated segmentation alternatives. (a) Original images. (b)-(e) The generated segmentation alternatives by thresholding the confidence map at 0.4, 0.5, 0.6, and 0.7, respectively. As can be seen, the generated segmentation alternatives contain at least a few acceptable segmentations. This shows that the confidence map obtained by integrating different cues is reliable. As a result, generating segmentations based on the confidence map is able to provide reasonable segmentation alternatives from which the user can choose one.

assess the performance of this editing method by comparing its segmentation results against the manual segmentations. The DSC and F-measure values for this method were 0.9670 and 0.9672 (precision = 0.9767; recall = 0.9578), respectively. Thus, generating alternative segmentations based on the confidence map is able to provide reasonable options for the user.

B. Interactive Editing through Graph Cuts

User interaction to correct erroneous segmentation results is key to providing accurate segmentations that meet practical needs. The interactive editing problem is posed as the Maximum A Posteriori (MAP) estimation of Markov Random Field (MRF). The MAP-MRF formulation minimizes a Gibbs energy function that is defined over the user input, presegmentation[1], and image intensity. Graph cuts are then used to obtain the optimal solution to the energy function.

1. Editing Energy Function

The goal of interactive editing is to obtain a new segmentation that satisfies a set of constraints: the user input, presegmentation, and image data. Interactive editing, analogous to image segmentation, is a labeling problem which involves assigning image pixels a set of labels [68]. Consider an image I containing a set of pixels $\mathcal{P} = \{1, 2, ..., M\}$, a set of labels $\mathcal{L} = \{l_i, l_2, ..., l_K\}$, the user input $\mathcal{U} = \{u_p : p \in \mathcal{P}, u_p \in \mathcal{L}\}$, and a nearly correct presegmentation $y = \{y_p : p \in \mathcal{P}, y_p \in \mathcal{L}\}$. The goal is to find a new optimal labeling x that minimizes the following energy

[1]Here, following the definition from [118], presegmentation is referred to as the prior, incorrect, pre-existing segmentation.

function [119]:

$$E\left(x \mid y,\mathcal{U},D\right)=\sum_{p\in\mathcal{P}}V_p\left(x_p \mid y,\mathcal{U},D\right)+\sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{N}_p}V_{pq}\left(x_p,x_q \mid D\right), \tag{4.2}$$

where $\mathcal{N}_p$ denotes the set of neighboring pixels of p, $V_p\left(x_p \mid y,\mathcal{U},D\right)$ is the unary clique potential, and $V_{pq}\left(x_p,x_q \mid D\right)$ is the piecewise clique potential. The user input and presegmentation are integrated into the unary potential, and image intensity is used in the piecewise potential to impose the boundary smoothness.

2. User Input and Presegmentation Constraints

The unary potential defines the weights between a node p and terminal nodes, s and t. The pixels the user indicates as the foreground or background serve as the hard constraints; the edge weights between these pixels and the terminals are set to infinity to prevent them from changing labels. Because a presegmentation is nearly correct, the new segmentation should be similar to the presegmentation after editing. Only the pixels with changed labels are penalized [97]. The penalty for changing a label is defined based on the intervening contour cue. The weights are:

$$w_{sp}=\begin{cases}\infty & p\in\mathcal{U}_f\\ \mathrm{IC}_b\left(p\right) & p\notin\mathcal{U}_f\end{cases} \tag{4.3}$$

and

$$w_{pt}=\begin{cases}\infty & p\in\mathcal{U}_b\\ \mathrm{IC}_f\left(p\right) & p\notin\mathcal{U}_b\end{cases}, \tag{4.4}$$

where $\mathcal{U}_f$ and $\mathcal{U}_b$ are the foreground and background labels, respectively, and $\mathrm{IC}_f\left(p\right)$ and $\mathrm{IC}_b\left(p\right)$ are the affinities between a pixel p to the nearest of the user labeled pixels $\mathcal{U}_f$ and $\mathcal{U}_b$, respectively. The unary potential is similar to that proposed by [118] that suggests use of the Euclidean distance from a pixel to the nearest of the user

labeld pixel; however, the work here considers the intervening contour cue that is important for distinguishing different objects. Figure 32 demonstrates the difference in using the Euclidean distance and the intervening contour cue as unary potential. As can be seen from Figure 32(d), the color clearly shows transitions from black (low penalty) to gray (high penalty) whereas no such transitions are shown in Figure 32(c) in which the penalties are given based on the Euclidean distance. The transitions in Figure 32(d) coincide with the object boundary, where a cut is more likely to occur.

3. Image Data Constraint

Piecewise potential ensures boundary smoothness by penalizing neighboring pixels assigned different labels. Based on Potts model [66], it is given as

$$V_{pq}\left(x_p, x_q \mid D\right) = w_{pq} \cdot \left(1 - \delta\left(x_p, x_q\right)\right) \ , \tag{4.5}$$

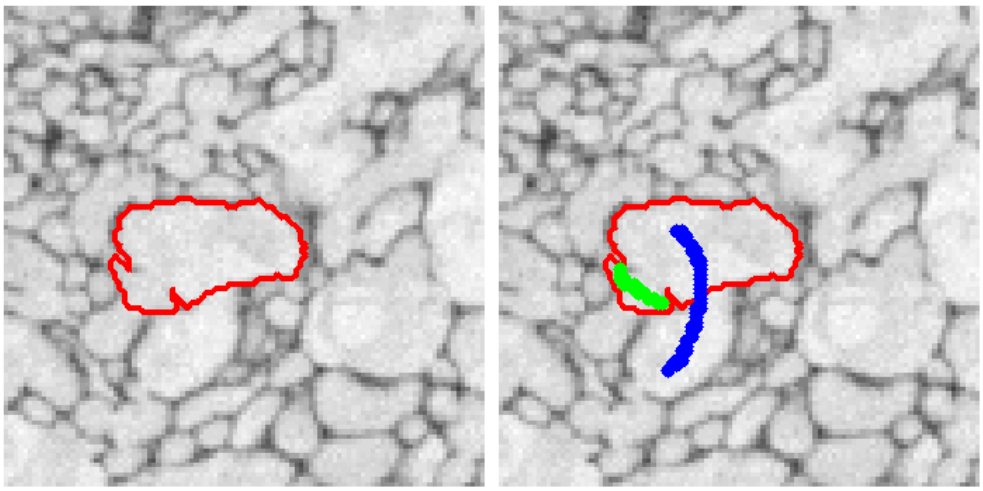
where $\delta\left(x_p, x_q\right)$ is the Kronecker delta defined as

$$\delta\left(x_p, x_q\right) = \begin{cases} 0 & \text{if } x_p \neq x_q \\ 1 & \text{if } x_p = x_q \end{cases} \ , \tag{4.6}$$

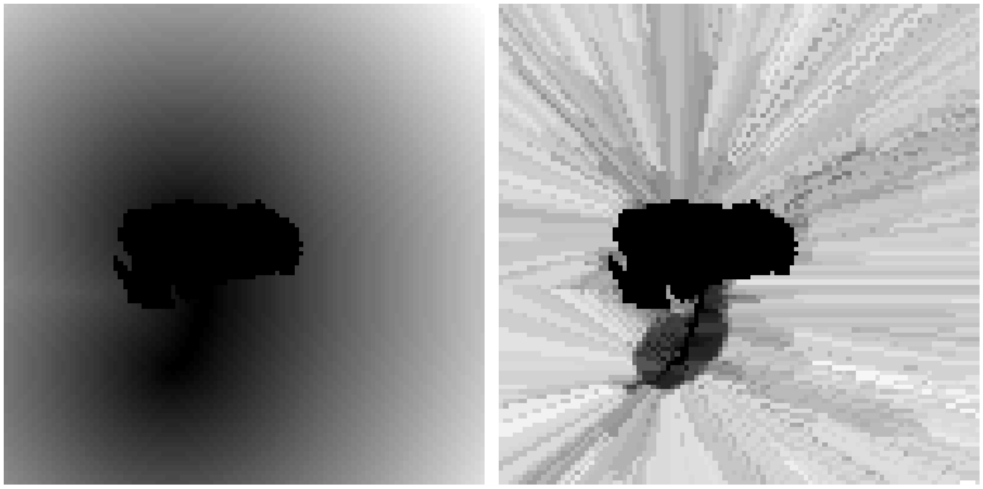
and w_{pq} is a penalty for assigning two neighboring pixels, p and q, different labels, defined using a Gaussian weighting function

$$w_{pq} = \exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma^2}\right) \cdot \frac{1}{\|p - q\|}, \tag{4.7}$$

where I_p and I_q are pixel intensities ranging from 0 to 255, $\|p - q\|$ is the Euclidean distance between p and q, and σ is a positive parameter set to 30.

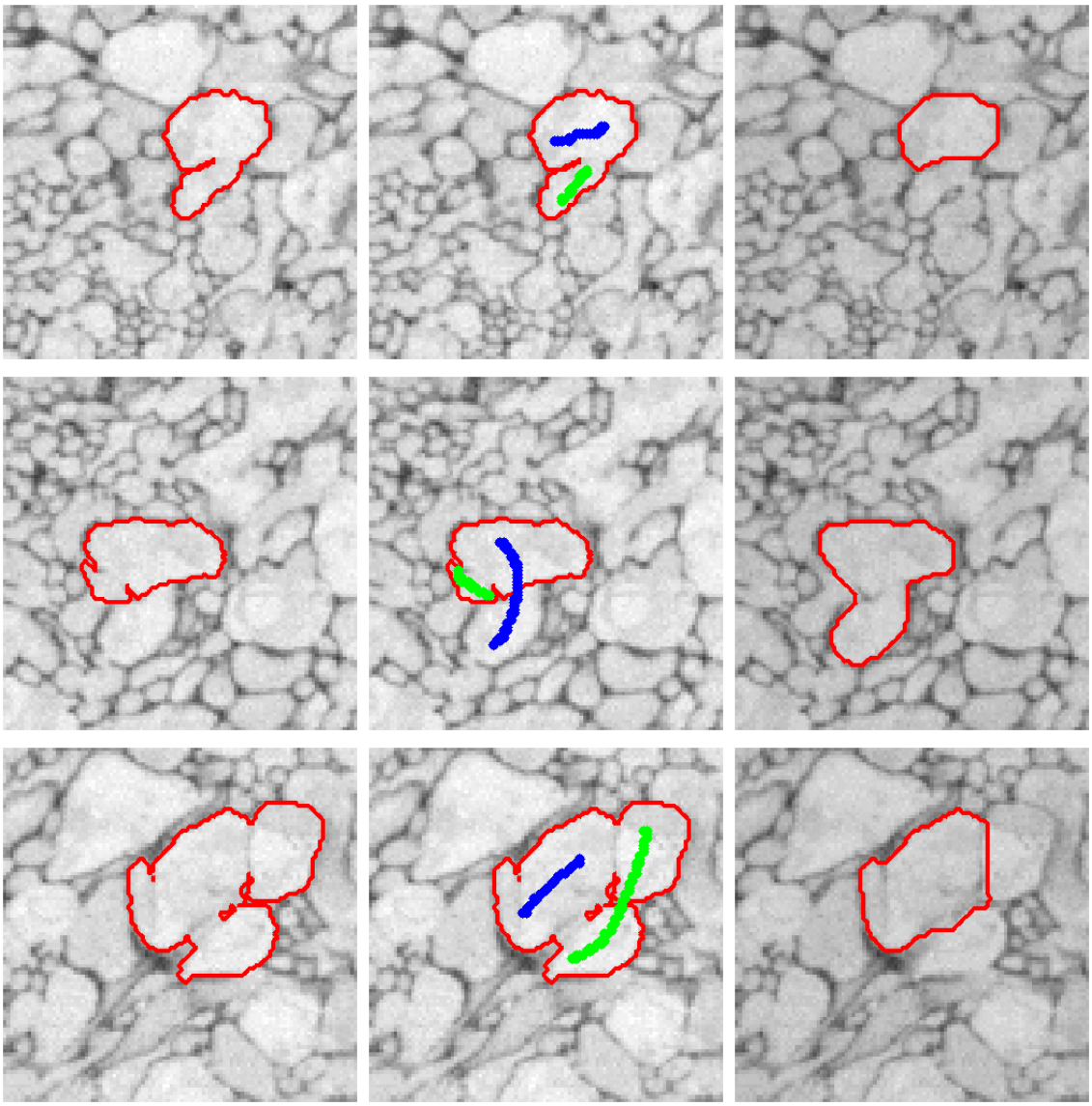


(a) (b)



(c) (d)

Fig. 32. Comparison of the difference between using the Euclidean distance and using the intervening contour cue in the unary potential. (a) Presegmentation. (b) User edit. The blue and green scribbles indicate the foreground and background marks, respectively. (c) The penalties, defined based on the Euclidean distance, for changing the label of pixels classified as background in (a) to foreground. (d) The penalties defined based on the intervening contour cue. Using the intervening contour cue in the unary potential, the color clearly shows transitions from black (low penalty) to gray (high penalty). The transitions coincide with the object boundary, where a cut is more likely to occur.



(a) Incorrect seg. (b) User edit (c) Result

Fig. 33. Examples of interactive segmentation editing. The editing algorithm takes a few of user inputs, together with the incorrect segmentation and image data constraints, and computes a new segmentation accordingly. The blue and green strokes represent the foreground and background marks, respectively.

4. Experimental Results

Similar to the experiments conducted for the editing through multiple choice method, 30 incorrect segmentations were used to evaluate the performance of the editing through graph cuts method. Figure 33 shows the examples of interactive editing through graph cuts. The first, second, and third columns of Figure 33 show the incorrect segmentations, user edits, and editing segmentation results. The blue and green strokes are the foreground and background marks, respectively. This editing tool gives the user flexibilities of placing strokes on a few pixels and produces a segmentation that meets the user's requirements.

For quantitative evaluation, Dice similarity coefficient (DSC) and F-measure were used to measure the segmentation accuracy of this editing method by comparing its segmentation results against the manual segmentations. The DSC and F-measure values obtained by this method were 0.9763 and 0.9810 (precision = 0.9676; recall = 0.9947), respectively, indicating that the editing results are highly overlapped with the manual segmentations.

C. Summary

This chapter presented an interactive editing framework that allows the user to correct segmentation errors produced by automated segmentation algorithms. By thresholding a confidence map using different values, the proposed editing framework first obtains a pool of alternative segmentations from which the user can select the most acceptable one, aiming for minimizing user interaction. In addition, the editing framework includes an editing tool that the user can place editing marks on a few pixels to produce the desired segmentation result. The editing task is formalized as an energy minimization problem and incorporates a set of constraints, ranging from the user

input and presegmentation to image data, into the energy function. Experimental results showed that the proposed editing framework provides a promising solution to the segmentation of SBFSEM image data sets.

CHAPTER V

TOPOLOGY-BASED VALIDATION

Manual editing can correct segmentation errors produced by automated segmentation algorithms, but it also introduces a practical challenge: the combination of multiple users' annotations of an image to obtain an estimation of the true, unknown labeling. Current estimation methods are not suited for electron microscopy (EM) images because they typically do not take into account topological correctness of a segmentation that can be critical in EM analysis. This chapter presents a topology-preserving segmentation ground truth estimation method for EM images from brain tissue. Taking a collection of annotations of an image, the algorithm aims to provide an estimated labeling that is topologically equivalent and geometrically similar to the true, unknown segmentation. Guided by the segmentation evaluation metric, warping error, the algorithm iteratively modifies topology of the estimated segmentation to maximize the topological agreements (i.e. minimize the disagreements) between the estimated segmentation and a set of given segmentations. By gradually changing its topology, the estimated segmentation becomes topologically equivalent to the true, unknown segmentation.

The section below details the evaluation metric, warping error, which measures topological errors of a segmentation against a reference labeling. Subsequently, the proposed method is presented.

A. Topological Disagreement Measure: the Warping Error

In supervised evaluation, the performance of a segmentation algorithm is quantitatively measured by comparing its segmentation results against a manually labeled ground truth (i.e. a reference image) based on evaluation metrics. Jaccard index [77],

Dice Similarity Coefficient (DSC) [78], and F-measure are well known and widely used metrics for segmentation evaluation. Those metrics use the amount of overlap between a segmentation and the ground truth as a similarity measure to evaluate the performance of a segmentation method. This makes them focus on measuring a segmentation's boundary accuracy at the pixel level but not take into account its topological correctness. However, in EM segmentation evaluation, measuring the degree of the topological correctness of a segmentation is also important because obtaining accurate analysis of the neural circuits relies on topologically correct reconstructions [30].

The warping error metric proposed by Jain et al. [30] is a metric that measures topological disagreements between segmentations and has been shown to be effective for EM segmentation evaluation. While comparing two segmentations, the error metric strongly penalizes topological disagreements but tolerates minor boundary localization differences. Conceptually, to calculate the topological disagreements between a segmentation and the ground truth, the ground truth image is first transformed into another image under topological and geometrical constraints, and then the disagreements (i.e. topological errors) can be identified as the pixel differences between the transformed image and the segmentation to be evaluated.

Before giving a formal definition of warping error, the concept of *warping* is first presented. Formally, given two binary images, L^* and L, if L^* can be transformed into L by flipping the labels of a sequence of pixels, L is called a *warping* of L^*, represented as $L \triangleleft L^*$. That is to say, L^* and L are topologically equivalent and geometrically similar. The labels of a sequence of pixels to be flipped are those of simple points (i.e. border pixels), where a point p is defined as a simple point if both the number of foreground connected components adjacent to p and the number of background connected components adjacent to p equal to 1. According to the theory of digital

Algorithm 1: A warping algorithm that warps a binary image L^* to a segmentation T. Algorithm from [30].

input : A binary image L^*, a segmentation T, and geometric constraints G

output: A warped image L

```
L = L*;
while true do
    S = simple(L) ∩ G ;
    i = argmax_{j∈S} |t_j − l_j|;
    if |t_i − l_i| > 0.5 then
        l_i = 1 − l_i;
    else
        return L;
```

topology, flipping the labels of simple points will not alter the object's topology.

Now, letting T be a segmentation to be evaluated and L^* be the reference annotation, the warping error $D(T \parallel L^*)$ is given as

$$D(T \parallel L^*) = \min_{L \triangleleft L^*} |E(L, T)| , \tag{5.1}$$

where L is the optimal warping of L^*, and $E(L, T)$ is the difference set (i.e. the pixels that have different labels in images L and T), defined as $E(L, T) = L \triangle T$. In other words, the warping error is considered as the pixel disagreements between the segmentation to be evaluated T and the transformed segmentation L. Note that in order to find minimal warping error, the image L^* is warped into L that is as similar to segmentation T as possible. The approach to warping a labeling L^* to another labeling L given T is detailed in Algorithm 1, and *simple(L)* indicates the

simple points of L. Figure 34 gives a detailed explanation of how the warping error is identified. First, the ground truth annotation in Figure 34(b) is warped into another labeling in Figure 34(c) by using the warping algorithm in Algorithm 1. Then, the topological disagreements presented in Figure 34(e) can be calculated from the pixel differences between the labeling in Figure 34(c) and the labeling in Figure 34(d), which contains two topological errors, a merge (blue circle) and a split (red circle). These errors result from the problem of boundary ambiguity in the original image, as can be seen in Figure 34(a).

B. Ground Truth Estimation by Maximizing Topological Agreements

This section focuses on the main contribution of this chapter: estimating a segmentation that is topologically equivalent and geometrically similar to the true, unknown segmentation when a few segmentations are available.

1. Problem Definition

Given a set of N segmentations, $S_1^*, ..., S_N^*$, either obtained by automated segmentation algorithms or annotated by different humans, the goal is to find an estimated segmentation $\hat{S}$ that is topologically equivalent and geometrically similar to the underlying unknown true segmentation. One potential segmentation that satisfies the topological and geometrical constraints and is capable of representing the true, unknown segmentation is that with a topology most of the given segmentations agree on. In other words, the estimated segmentation $\hat{S}$ is a segmentation that minimizes the warping error between itself and the given segmentations. Mathematically, $\hat{S}$ is obtained by minimizing the following:

$$\hat{S} = \underset{S}{\operatorname{argmin}} \sum_{i=1}^{N} D\left(S \parallel S_i^*\right) = \underset{S}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{S_i \lhd S_i^*} \left| E\left(S, S_i\right) \right| , \tag{5.2}$$

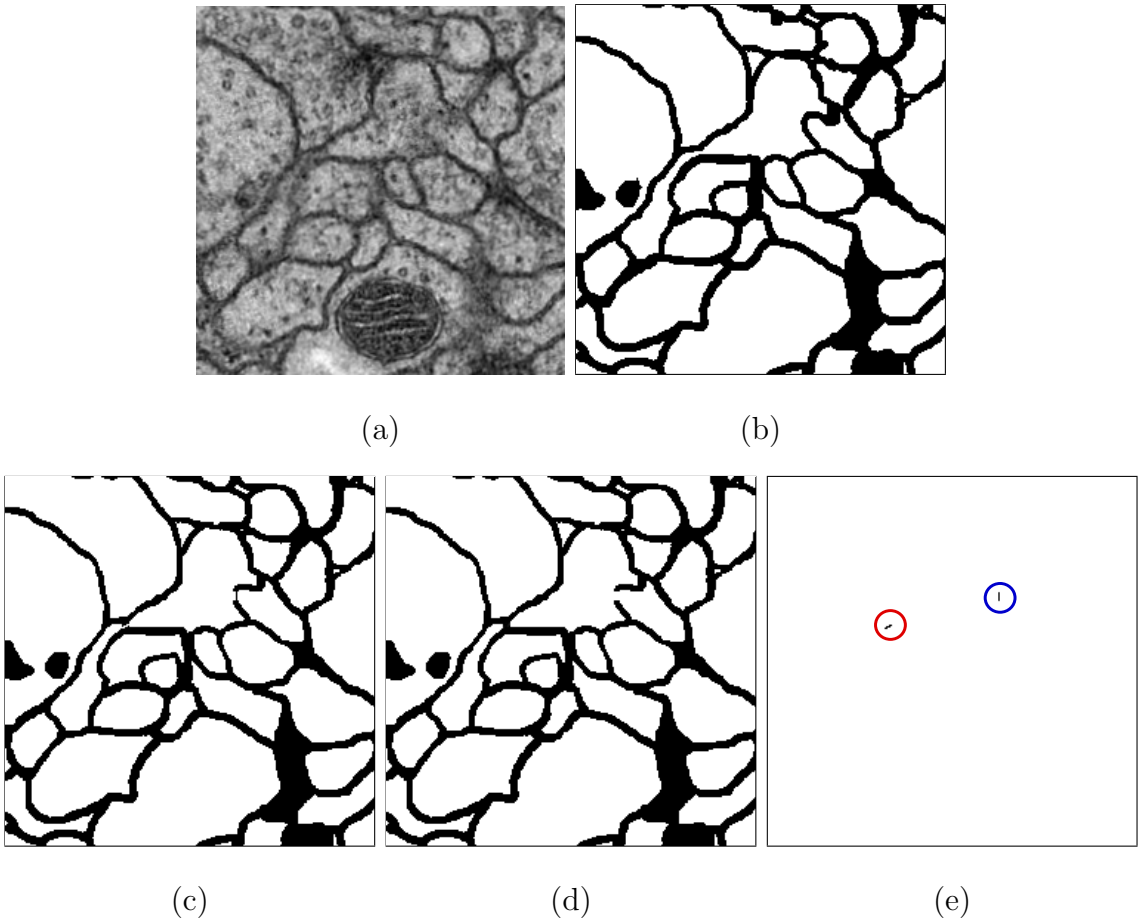


Fig. 34. Measured warping error of a segmentation against the ground truth. (a) A sample EM image. (b) Manually annotated ground truth. (c) A warping of the ground truth image shown in (b) given a segmentation in (d). (d) A segmentation to be evaluated. (e) Measured warping error. The pixel disagreements between (c) and (d) consist of the topological errors. In this case, two topological errors, a merge (blue circle) and a split (red circle), occur due to the boundary ambiguity in the original image.

where S_i is the optimal warping of the labeling S_i^*.

One possible method to find the estimated segmentation $\hat{S}$ is to enumerate all possible labelings and choose one that has minimal warping error. However, enumerating all labelings can be computationally expensive. Another alternative is to gradually change the topology of a segmentation and make it converge to a topology that most segmentations agree on. The section below details this approach.

2. Topological Correction Algorithm

Changing the labeling of an image involves flipping the labels of pixels, which can result in a merger of two adjacent regions or a splitting of a region into two. The potential pixels whose change of label causes a topological change are those that affect warping error. To achieve the goal of seeking an estimated segmentation with a topology that most segmentations agree on, the algorithm starts with an initial segmentation obtained by using the majority voting method. At each iteration, by using the number of topological errors as the evaluation metric, the algorithm corrects one topological disagreement between the estimated segmentation and the given segmentations. While correcting a topological disagreement at each iteration, the algorithm selects an error having a lowest flipping cost defined in Equation 5.4, detailed in the next section. A new labeling is accepted only if it has less warping error. The algorithm repeats the process of correcting topological errors and stops when no topological changes can lead to the reduction in the overall warping error, that is, it reaches a segmentation that minimizes warping error defined in Equation 5.2. Algorithm 2 details this proposed method.

Algorithm 2: Topological correction by minimizing warping error (proposed algorithm).

input : A set of labled binary images, $S_1^*, ..., S_N^*$

output: An estimate of the ground truth, $\hat{S}$

initialize $\hat{S}$ to the result of the majority voting method given $S_1^*, ..., S_N^*$;

foreach S_i^* **do**

  $E_i = D(\hat{S} \parallel S_i^*)$;

$E_{\min} = \sum_i E_i$;

while *not converged* **do**

  assign each topological error the flipping cost based on Equation 5.4;

  select a topological error with the lowest flipping cost;

  flip the selected pixels in the estimated segmentation $\hat{S}$;

  foreach S_i^* **do**

    $E_i = D(\hat{S} \parallel S_i^*)$;

  $E_{\text{new}} = \sum_i E_i$;

  if $E_{\text{new}} < E_{\min}$ **then**

    accept the new estimated ground truth $\hat{S}$;

  else

    reject the new estimated ground truth and restore $\hat{S}$ back to the previous estimation;

3. Topological Change Cost

As mentioned above, flipping the label of pixels that contain warping error modifies the topology of a segmentation. To better locate the pixels for topological change, each pixel is associated with a flipping cost, and the selection of what pixels' labels to be changed depends on the cost associated with those pixels. More specifically, the flipping cost of each pixel contains statistical information of an image, such as the intensity distributions.

To define the flipping cost, two notations are first introduced. Let S be the foreground (object) segmentation and $\bar{S}$ be the background segmentation. The cost of flipping the label of a pixel p from S_p to $\bar{S}_p$, $f(p)$, is defined as

$$f(p) = \frac{\Pr(I_p|S)}{\Pr\left(I_p|\bar{S}\right) + \Pr(I_p|S)}, \tag{5.3}$$

where I_p is the intensity value of pixel p, and $\Pr(I_p|S)$ and $\Pr\left(I_p|\bar{S}\right)$ represent how well the intensity of pixel p fits into the intensity distributions (histograms) of foreground and background, respectively. Because a set of segmentations are given, the intensity histograms for the foreground and background are available. Figure 35 illustrates how $\Pr(I_p|S)$ and $\Pr\left(I_p|\bar{S}\right)$ are calculated, where blue and red curves are the probability of a point belonging to the foreground and background, respectively. Figure 36(a) shows the flipping cost of changing a label of each pixel from the foreground to the background. Brighter color indicates a higher cost. Similarly, the cost of changing the label of a pixel p from $\bar{S}_p$ to S_p is defined by $\Pr\left(I_p|\bar{S}\right) / \left(\Pr\left(I_p|\bar{S}\right) + \Pr(I_p|S)\right)$. The flipping cost of changing a label of each pixel from the background to the foreground is shown in Figure 36(b).

As can be seen in the example given in Figure 34, the topological change of a segmentation requires a sequence of pixel flips. Now, let C denote a set of pixels

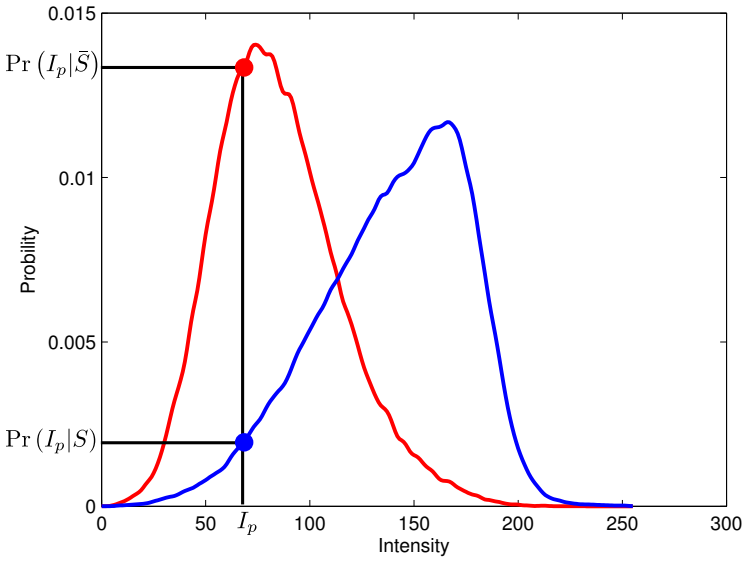
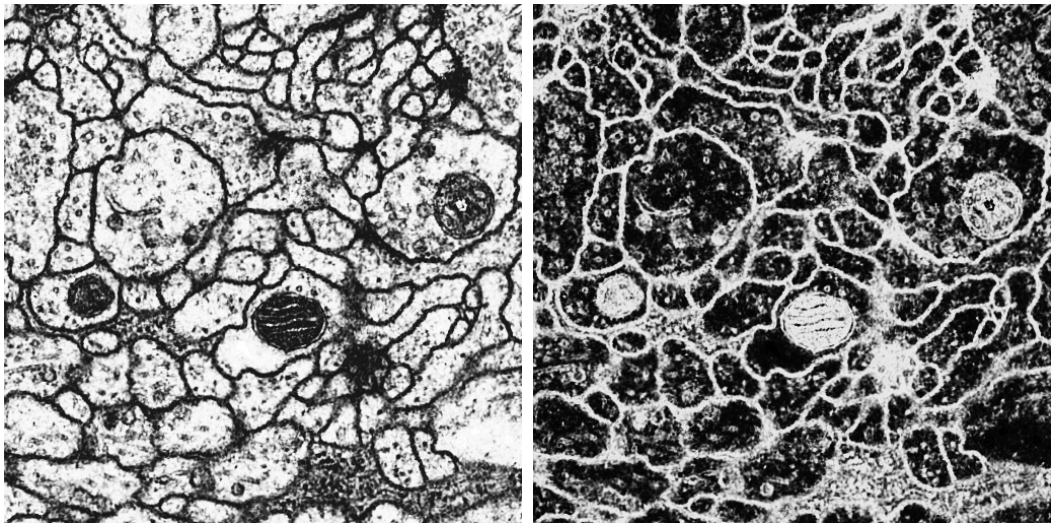


Fig. 35. Probability of a point belonging to foreground (S) and background ($\bar{S}$) based on the given foreground and background intensity distributions. Red and blue curves are the probability of a point belonging to the background and foreground, respectively. I_p denotes the intensity of a point p, and $\Pr(I_p|S)$ and $\Pr(I_p|\bar{S})$ denote the probabilities.

involved in the merger of two adjacent regions or the splitting of a region. To reduce the computational complexity associated with calculating the flipping cost of the pixels, a simple assumption is made that the pixel flip is independent with each other. Therefore, the cost $f(C)$ of flipping all points in C, is defined as sum of the flipping cost of the individual pixels, that is,

$$f(C) = \sum_{p \in C} f(p) \ . \tag{5.4}$$



(a) (b)

Fig. 36. Flipping cost of changing a label of each point in an image. (a) The flipping cost of changing a label of each pixel from the foreground to the background. (b) The flipping cost of changing a label of each pixel from the background to the foreground. Brighter color indicates a higher cost.

C. Experimental Results

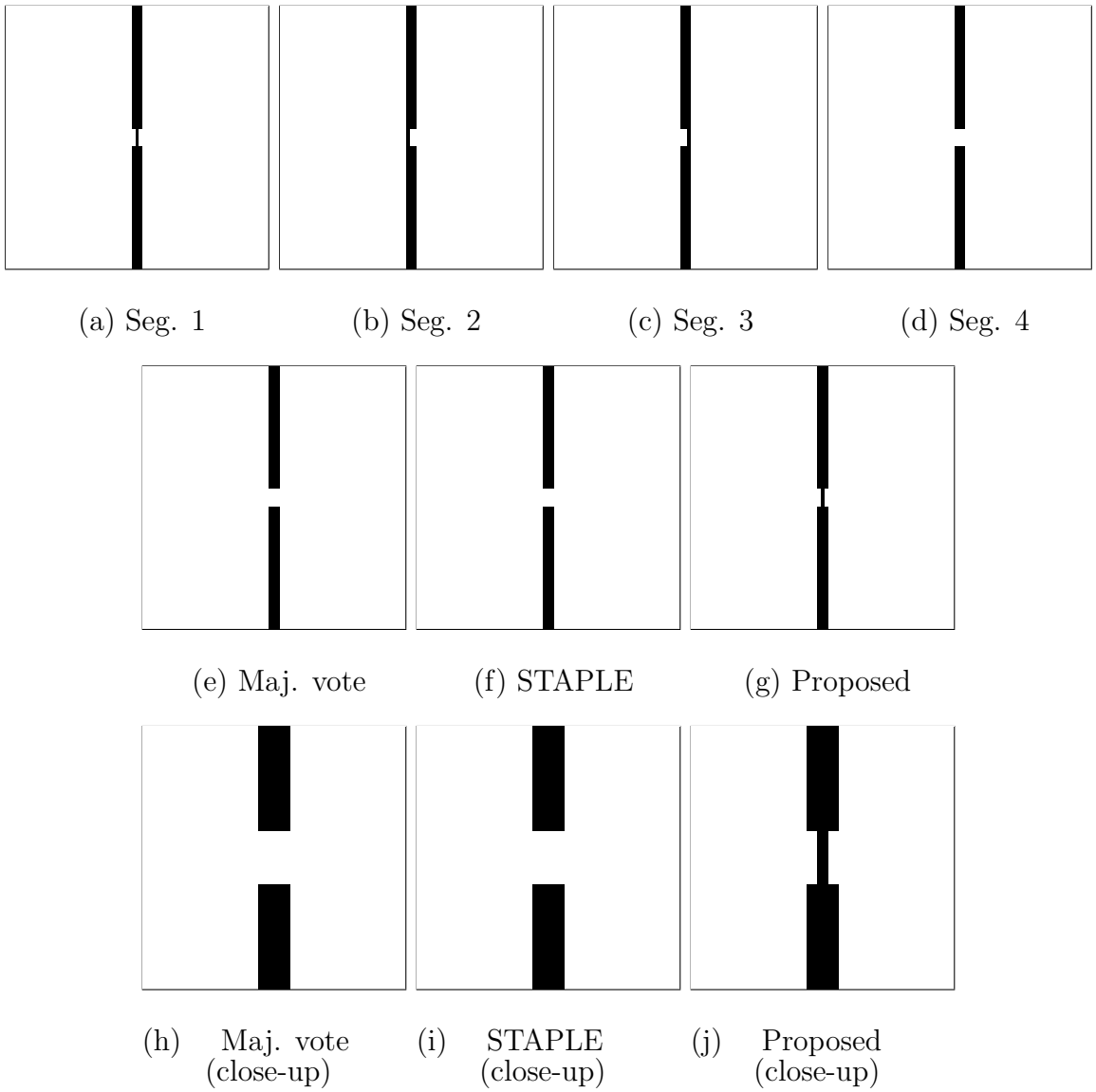
The experiments were carried out on a few synthetic images and on an EM data set [24, 25]. The purpose of applying the proposed method to simple synthetic images was to show that the proposed method can retrieve a segmentation that is topologically equivalent to the true, unknown segmentation.

1. Synthetic Images

A set of segmentations of an image are required to evaluate the proposed method. Four alternative segmentations were generated, and they are shown in Figures 37(a) to 37(d). The first three segmentations separate the image into two regions (i.e. same topology) with a slightly different boundary localization while the last (Figure 37(d))

segments the image as a whole region (different topology from the rest). These four segmentations present four possible segmentations of an image, and they are to be combined to obtain an estimate of the underlying true segmentation.

Demonstrated in Figure 37 is a simple comparison between the estimated segmentation obtained by the proposed method and those by majority voting and STAPLE. The four alternative segmentations are high quality, approaching expert levels: The estimated sensitivities (i.e. the probability of an annotator labeling a pixel as foreground if the true label is foreground) were 1.0000, 1.0000, 1.0000, and 1.0000, respectively, and the estimated specificities (i.e. the probability of an annotator labeling a pixel as background if the true label is background) were 0.9556, 0.9556, 0.9556, and 0.9333, respectively. The estimated segmentation of the true, unknown segmentation by using the majority voting method is shown in Figure 37(e), which is a segmentation that most segmentations agree on. However, this estimated segmentation is unable to represent the true, unknown segmentation because they are not topologically equivalent. Different from the majority voting method that treats each segmentation equally, STAPLE weights individual segmentation depending on its estimated performance level. The combined estimated result by using STAPLE is shown in Figure 37(f). However, we can see that although most of the alternative segmentations are topologically correct, such as those in Figures 37(a) to 37(c), STAPLE is unable to produce a topologically correct estimate of the true, unknown segmentation. This indicates its estimation is sensitive to the boundary localization of given labelings. On the other hand, when the same set of initial segmentations were given, our proposed algorithm produced a topologically correct estimate. The resulting segmentation is shown in Figure 37(g), and its respective close-up in Figure 37(j).



(a) Seg. 1 (b) Seg. 2 (c) Seg. 3 (d) Seg. 4

(e) Maj. vote (f) STAPLE (g) Proposed

(h) Maj. vote (close-up) (i) STAPLE (close-up) (j) Proposed (close-up)

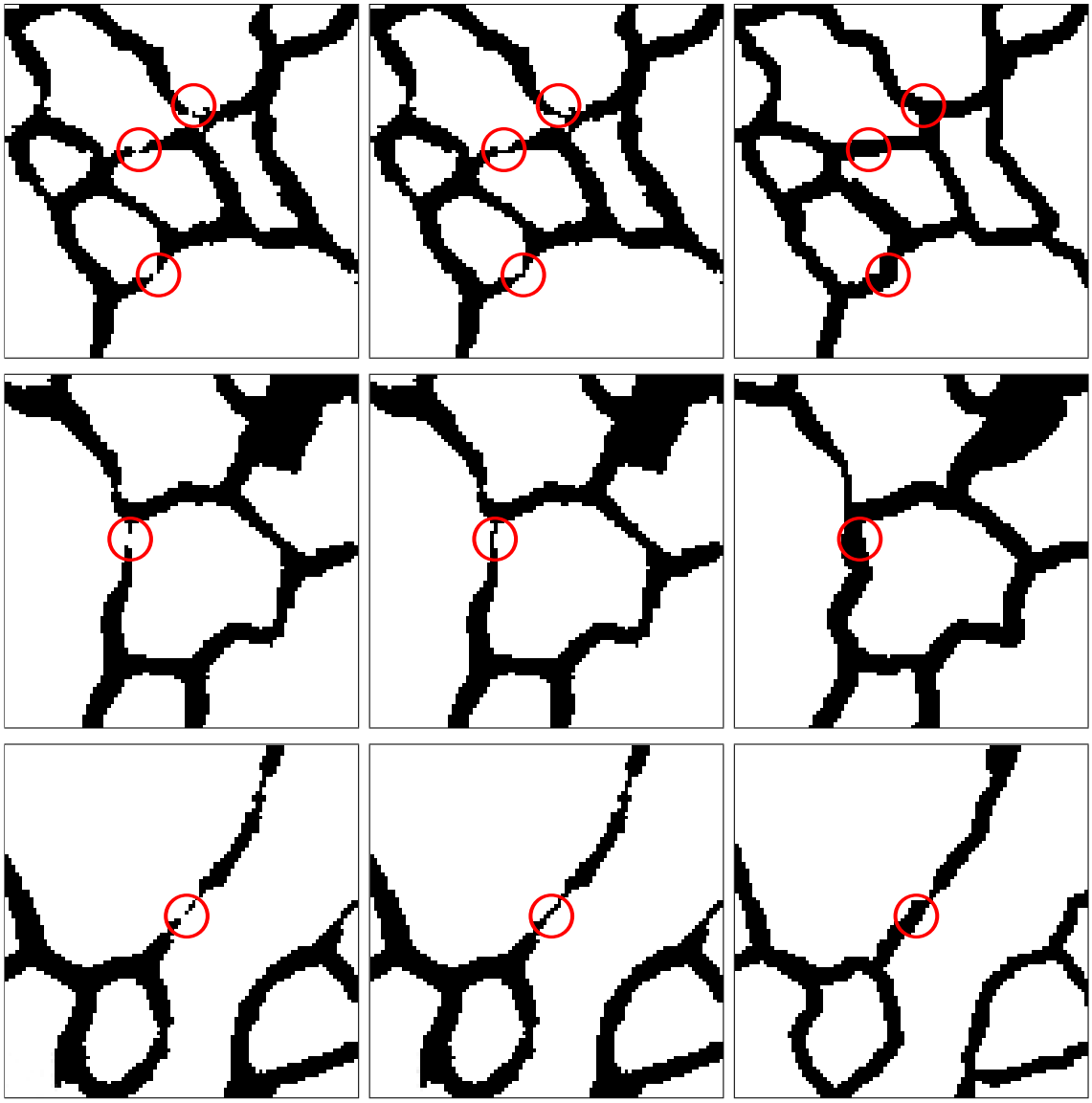
Fig. 37. Comparison between the estimated segmentation obtained by the proposed method and those by the majority voting and STAPLE. (a)-(d) Alternative segmentations to be fused. The first three segmentations separate the image into two regions by a slightly different boundary localization. The last segments the image as a whole region. (e) The estimated segmentation by majority voting merges two separate regions as one, thus causing a merge error. (f) The estimated segmentation by STAPLE also merges two separate regions as one, thus causing a merge error. (g) The estimated segmentation obtained by the proposed method is topologically correct. (h)-(j) Close-ups of the results from majority voting, STAPLE, and the proposed algorithm, respectively.

2. EM Data

A serial section Transmission Electron Microscopy (ssTEM) data set of the Drosophila first instar larva ventral nerve cord (VNC) from Cardona et al. [24, 25] was used for the evaluation of the proposed method. The data set was manually delineated by an expert, and the manual segmentations served as the ground truth the algorithm aims to estimate.

To test the developed method, a number of segmentations were first generated by thresholding the image at different values with additional manual editing to finally construct alternative segmentations. Note that the main focus of the work is not on how these alternative segmentations are generated, so any reasonable manual or automated method will be enough. These generated segmentations represent the alternative segmentations to be combined. Taking those segmentations as input, the proposed method produced an estimated segmentation. Figure 38(a) shows the initial segmentations with which the proposed method starts (majority vote), Figure 38(b) the estimated segmentations obtained by the proposed method, and Figure 38(c) the ground truth annotated by the expert. The topologies of the initial segmentations, obtained by the majority voting method, do not agree with those of the ground truth, which are indicated by the red circles. The final estimated segmentations, on the other hand, are topologically equivalent and geometrically similar to the ground truth, with a minor boundary localization difference.

Using the same set of input segmentations and warping error as the evaluation metric, the quantitative comparison of the results obtained by the majority voting method, STAPLE, and the proposed method is shown in Table VII. As we can see, topological errors exist in the results obtained by majority voting and STAPLE because these two methods fuse segmentation labels at the pixel level. The proposed



(a) Initial (Majority vote) (b) Estimated (c) Ground truth

Fig. 38. Comparison of the topologies of initial segmentations, estimated segmentations, and ground truth. (a) The initial estimated segmentations with which the validation method starts. (b) The estimated segmentations produced by the validation method. (c) Ground truth. The topologies of the initial segmentations (a) do not agree with those of the ground truth (c), which are indicated by the red circles. The estimated segmentations (b) produced by the proposed method, on the other hand, are topologically equivalent and geometrically similar to the ground truth.

Table VII. Comparison of the number of topological errors committed by the majority voting method, STAPLE, and the proposed method on 10 different samples from the EM data set. As we can see, topological errors exist in the results obtained by the majority voting method and STAPLE whereas the proposed method is able to obtain topologically correct segmentations as long as topologies of most of the alternative segmentations are correct.

Sample #	1	2	3	4	5	6	7	8	9	10
Majority voting method	6	4	8	2	2	2	0	2	6	5
STAPLE	2	4	3	5	2	0	3	2	3	2
Proposed method	0	0	0	0	0	0	0	0	0	0

method, on the contrary, can obtain topologically correct segmentations as long as the topologies of majority of alternative segmentations are correct. Also note that, in the experiment, the proposed method used the majority voting method's results as the initial estimated segmentations and gradually modified the topologies of the estimated segmentations until convergence. The proposed method's final estimated segmentations are topologically equivalent to the true segmentations even it started with segmentations containing topological errors.

D. Summary

This chapter presented a novel pooling method that seeks a segmentation topologically equivalent and geometrical similar to the true, unknown segmentation when a set of alternative segmentations are available. This method is effective for noisy EM images because it maximizes the topological agreements among segmentations during the estimation process and ensures that a segmentation is topologically correct, which is important for connection estimation for connectomics research. Experimental re-

sults have demonstrated the effectiveness of this method.

CHAPTER VI

DISCUSSION

This chapter starts with presenting the contributions of this dissertation, followed by discussion, and finally offers several potential future research directions for improving the developed algorithms.

A. Contributions

The main contribution of this dissertation is to provide a thorough framework for reconstruction of neuronal morphology from stacks of serial EM images. This includes segmentation algorithms for obtaining structural anatomy of neural circuits, interactive segmentation tools for manual correction of erroneous segmentations, and a validation method for obtaining a topologically correct segmentation.

In this dissertation, a set of novel image segmentation algorithms for EM reconstruction were first presented. The contribution lies in the utilization of geometrical information (i.e. shape prior) available from adjacent images in 3D for the segmentation task. The geometrical information serves as a constraint in the segmentation process, and the incorporation of such information into an energy function greatly improves segmentation accuracy, especially for SBFSEM imaging data, which can be seen from Figure 13. Besides, creation of labeled data sets can be very laborious. The proposed method needs no such creation of human labeled data that is required for a machine learning method while obtaining comparable performance. The developed segmentation algorithms have yielded success in EM reconstruction, which was demonstrated in Chapter III.

Second, interactive editing tools that allow the user to easily and quickly refine incorrect segmentations were also provided. These tools aim to minimize the amount

of effort required for manual correction through the two following aspects: (1) Editing through multiple choice. This tool integrates multiple cues to provide a set of segmentation alternatives from which the user can select the most acceptable one, if available. (2) Interactive editing through graph cuts. This tool takes user inputs and accordingly produces a segmentation that satisfies the user's constraints as well as underlying image constraints.

Finally, a new segmentation validation method was proposed. When a set of segmentations are available, the validation method seeks a segmentation with a topology that most segmentations agree on (i.e. the majority voting based on the topology). It uses warping error as a measure of topological agreements among segmentations and minimizes this error to ensure that the reconstructions are topologically correct. Obtaining topologically correct reconstructions is crucial for subsequent connectivity analysis of the 3D anatomical structures.

B. Limitations

Although the proposed algorithms have successfully reconstructed anatomical structures from stacks of serial EM images, several open issues or problems remain to be solved.

The developed tracking and segmentation algorithm requires the user to delineate initial neuronal membranes with which the algorithm starts. To speed up reconstruction and reduce the amount of user interference, a method for automatically providing initial delineations for the tracking algorithm should be investigated.

Another limitation of the tracking algorithm is that when an error occurs in a slice, this error accumulates in the subsequent slices, which is a common problem in most of the tracking algorithms. Currently, correction of these errors mainly relies on

manual methods. To make the algorithm more robust, finding solutions to alleviating this problem is required.

One strength of the proposed segmentation algorithms is the utilization of geometrical information available from adjacent images to constrain the segmentation process when the problem of boundary ambiguity occurs. However, how to properly weight this geometrical information term in the energy function is an issue. Though using the curvedness to adaptively adjust the weighting is considered in the proposed algorithms, it is still not perfect due to uneven illumination and noise inherent in EM data. The influence of this geometrical information is more emphasized in the ssTEM data because the z resolution (section thickness) is much lower than the x-y resolution, and the shapes of cells sometimes change dramatically.

The editing through multiple choice method provides segmentation alternatives from which the user can select the most acceptable one. This alleviates the human labor required for editing. One potential direction to improve this scheme is to provide a confidence value for each alternative segmentation such that the user can select a segmentation based on the given confidence measure of segmentations. Also, such confidence values can be used by the user to quickly locate regions that need correction.

For the topology-based validation method, it uses the warping error as a topological measurement and thus ensures that an estimated segmentation is topologically equivalent to the true, unknown labeling. Currently, the proposed method is for 2D validation. To make it more general, the method of ensuring that the 3D reconstruction is topologically correct should be explored.

C. Future Research Directions

Based on the open issues and problems discussed in the above section, several future directions are possible for improving the performance of the developed algorithms, including the following aspects.

Utilization of More Contextual Information. Contextual information has been proven to be an important cue that significantly improves performance of algorithms proposed for many tasks in the domain of natural image analysis, such as detection [120–122], classification [123], and segmentation [123, 124]. Analogously, contextual information is a helpful cue for EM segmentation, which has been demonstrated in Chapter III. The proposed algorithms use such information available from adjacent images to resolve the problem of boundary ambiguity. In fact, in 3D volumetric imaging data, rich contextual information is available when the field of view is increased to include more images in the z direction or more neighboring objects (regions) in the x-y plane.

Including more contextual information requires the learning of an efficient model to represent the underlying context, a topic that remains mostly unsolved. In some vision models, such as Markov random fields (MRF) and conditional random fields (CRF), only short range contextual information are captured between a node and its neighbors. How to incorporate a larger range of contextual information in a model is a promising topic for future research.

Automated Correction of Segmentation Errors. With the availability of more EM imaging data and more reconstruction methods developed for the analysis of EM data, more robust validation methods for proofreading the reconstruction results are highly necessary. Currently, manual proofreading is a commonly used approach,

but it is time-consuming and arduous. Hence, automatic correction methods that autonomously correct reconstruction errors will provide an ideal solution to reducing the amount of manual interaction required for the connectomic reconstruction. To this end, automatic correction algorithms should be equipped with the two following properties:

- Using the knowledge of neuronal topologies to analyze the reconstruction results and to detect potential errors.
- Capability to automatically correct segmentation errors. When errors are found in a reconstruction, a correction algorithm should provide potential corrections.

A correction method equipped with the ability to correct segmentation errors will not only eliminate the aforementioned problem of error accumulation in the tracking algorithm but also provide more reliable reconstructions.

3D Topological Validation A natural extension of the proposed 2D validation method presented in Chapter V is to extend it to 3D. Similar to the 2D case where the proposed method takes a set of segmentations as input, a 3D validation method will take a few 3D reconstructions (obtained by different segmentation algorithms or by the same segmentation method with different parameter settings) as input and seek a 3D reconstruction with a topology that most reconstructions agree on. Warping error could be used as the evaluation metric for evaluating the topological agreement between two 3D reconstructions. This 3D validation method will greatly facilitate connection estimation for connectomics research.

CHAPTER VII

CONCLUSION

With advances in the development of EM imaging techniques, vast data sets containing numerous anatomical structures have been produced in the past few years. These data sets are key to the reconstruction of complete neural circuits of the nervous system (i.e. the connectome) and to the elucidation of how the brain functions. However, these imaging data also pose significant challenges to reconstruction. For this, in this dissertation, a set of methods, from segmentation methods and interactive editing to validation methods, have been developed to facilitate the reconstruction task. Each of these algorithms plays a unique part in the process of reconstruction: the segmentation methods extract 3D anatomical structures from stacks of serial EM images, interactive editing provides an efficient way to correct segmentation errors, and finally the topology-based validation method ensures that the reconstruction is topologically correct.

More effort is needed to achieve the grand goal of mapping the complete wiring diagram of the nervous system. I expect the algorithms presented in this dissertation to contribute to the reconstruction of the connectome and to open new directions in the development of reconstruction methods.

REFERENCES

[1] K. L. Briggman and W. Denk, "Towards neural circuit reconstruction with volume electron microscopy techniques," *Current Opinion in Neurobiology*, vol. 16, no. 5, pp. 562–570, 2006.

[2] D. A. Drachman, "Do we have brain to spare?" *Neurology*, vol. 64, no. 12, pp. 2004–5, 2005.

[3] J. W. Lichtman, J. Livet, and J. R. Sanes, "A technicolour approach to the connectome," *Nature Reviews Neuroscience*, vol. 9, no. 6, pp. 417–422, 2008.

[4] J. W. Lichtman and J. R. Sanes, "Ome sweet ome: What can the genome tell us about the connectome?" *Current Opinion in Neurobiology*, vol. 18, no. 3, pp. 346–353, 2008.

[5] O. Sporns, G. Tononi, and R. Kötter, "The human connectome: A structural description of the human brain," *PLoS Comput Biol*, vol. 1, no. 4, p. e42, 2005.

[6] B. L. Chen, D. H. Hall, and D. B. Chklovskii, "Wiring optimization can relate neuronal structure and function," vol. 103, no. 12, pp. 4723–4728, 2006.

[7] A.-S. Chiang, C.-Y. Lin, C.-C. Chuang, H.-M. Chang, C.-H. Hsieh, C.-W. Yeh, C.-T. Shih, J.-J. Wu, G.-T. Wang, Y.-C. Chen, C.-C. Wu, G.-Y. Chen, Y.-T. Ching, P.-C. Lee, C.-Y. Lin, H.-H. Lin, C.-C. Wu, H.-W. Hsu, Y.-A. Huang, J.-Y. Chen, H.-J. Chiang, C.-F. Lu, R.-F. Ni, C.-Y. Yeh, and J.-K. Hwang, "Three-dimensional reconstruction of brain-wide wiring networks in Drosophila at single-cell resolution," *Current Biology*, vol. 21, no. 5, pp. 1–11, 2011.

[8] J.-J. Fernandez, C. O. S. Sorzano, R. Marabini, and J.-M. Carazo, "Image processing and 3-D reconstruction in electron microscopy," *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 84–94, 2006.

[9] Y. Mishchenko, T. Hu, J. Spacek, J. Mendenhall, K. M. Harris, and D. B. Chklovskii, "Ultrastructural analysis of hippocampal neuropil from the connectomics perspective," *Neuron*, vol. 67, no. 6, pp. 1009–1020, 2010.

[10] W. Denk and H. Horstmann, "Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure," *PLoS Biology*, vol. 2, no. 11, p. e329, 2004.

[11] K. M. Harris, E. Perry, J. Bourne, M. Feinberg, L. Ostroff, and J. Hurlburt, "Uniform serial sectioning for transmission electron microscopy," *J. Neurosci.*, vol. 26, no. 47, pp. 12 101–12 103, 2006.

[12] M. Helmstaedter, K. L. Briggman, and W. Denk, "3D structural imaging of the brain with photons and electrons," *Current Opinion in Neurobiology*, vol. 18, no. 6, pp. 633–641, 2008.

[13] J. Livet, T. A. Weissman, H. Kang, R. W. Draft, J. Lu, R. A. Bennis, J. R. Sanes, and J. W. Lichtman, "Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system," *Nature*, vol. 450, pp. 56–62, 2007.

[14] B. H. McCormick, "The knife-edge scanning microscope," Department of Computer Science, Texas A&M University, Tech. Rep., 2003.

[15] K. D. Micheva and S. J. Smith, "Array tomography: A new tool for imaging the molecular architecture and ultrastructure of neural circuits," *Neuron*, vol. 55, no. 1, pp. 25–36, 2007.

[16] S. J. Smith, "Circuit reconstruction tools today," *Current Opinion in Neurobiology*, vol. 17, no. 5, pp. 601–608, 2007.

[17] P. S. Tsai, B. Friedman, A. I. Ifarraguerri, B. D. Thompson, V. Lev-Ram, C. B. Schaffer, Q. Xiong, R. Y. Tsien, J. A. Squier, and D. Kleinfeld, "All-optical histology using ultrashort laser pulses," *Neuron*, vol. 39, no. 1, pp. 27–41, 2003.

[18] D. Mayerich, L. Abbott, and B. McCormick, "Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain," *Journal of Microscopy*, vol. 231, no. 1, pp. 134–143, 2008.

[19] B. H. McCormick, U.S. Patent 6 744 572, 2004.

[20] E. R. Macagno, C. Levinthal, and I. Sobel, "Three-dimensional computer reconstruction of neurons and neuronal assemblies," *Annual Review of Biophysics and Bioengineering*, vol. 8, pp. 323–351, 1979.

[21] J. K. Stevens, T. L. Davis, N. Friedman, and P. Sterling, "A systematic approach to reconstructing microcircuitry by electron microscopy of serial sections," *Brain Research Reviews*, vol. 2, no. 1-3, pp. 265–293, 1980.

[22] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode caenorhabditis elegans," *Philosophical Trans. Royal Society of London - Series B: Biological Sciences*, vol. 314, pp. 1–340, 1986.

[23] K. J. Hayworth, N. Kasthuri, R. Schalek, and J. W. Lichtman, "Automating the collection of ultrathin serial sections for large volume TEM reconstructions," *Microscopy and Microanalysis*, vol. 12, pp. 86–87, 2006.

[24] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein, "An integrated micro- and macroarchitectural analysis of the *Drosophila* brain by computer-assisted serial section electron microscopy," *PLoS Biol*, vol. 8, no. 10, p. e1000502, 2010.

[25] A. Cardona, "Segmented ssTEM stack of neural tissue," http://www.ini.uzh.ch/~acardona/data.html, 2010.

[26] D. B. Chklovskii, S. N. Vitaladevuni, and L. K. Scheffer, "Semi-automated reconstruction of neural circuits using electron microscopy," *Current Opinion in Neurobiology*, vol. 20, no. 5, pp. 667–675, 2010.

[27] R. Kumar, A. Vázquez-Reina, and H. Pfister, "Radon-like features and their application to connectomics," in *IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis*, 2010.

[28] Y. Mishchenko, "Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs," *J. Neuroscience Methods*, vol. 176, no. 2, pp. 276–289, 2009.

[29] V. Jain, J. F. Murray, F. Roth, S. C. Turaga, V. P. Zhigulin, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung, "Supervised learning of image restoration with convolutional networks," in *Proc. IEEE Int'l Conf. Computer Vision*, 2007, pp. 1–8.

[30] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. N. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, K. M. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. C. Tapia, J. W. Lichtman, and H. S. Seung, "Boundary learning by optimization with topological

constraints," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2488–2495.

[31] S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung, "Maximin affinity learning of image segmentation," in *Advances in Neural Information Processing Systems*, 2009, pp. 1865–1873.

[32] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. L. Briggman, W. Denk, and H. S. Seung, "Convolutional networks can learn to generate affinity graphs for image segmentation," *Neural Computation*, vol. 22, no. 2, pp. 511–538, 2010.

[33] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F. A. Hamprecht, "Segmentation of SBFSEM volume data of neural tissue by hierarchical classification," in *Proc. DAGM Symp.*, 2008, pp. 142–152.

[34] V. Kaynig, T. Fuchs, and J. M. Buhmann, "Neuron geometry extraction by perceptual grouping in ssTEM images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2902–2909.

[35] E. Jurrus, A. R. C. Paiva, S. Watanabe, J. R. Anderson, B. W. Jones, R. T. Whitaker, E. M. Jorgensen, R. E. Marc, and T. Tasdizen, "Detection of neuron membranes in electron microscopy images using a serial neural network architecture," *Medical Image Analysis*, vol. 14, no. 6, pp. 770–783, 2010.

[36] K. U. Venkataraju, A. R. C. Paiva, E. Jurrus, and T. Tasdizen, "Automatic markup of neural cell membranes using boosted decision stumps," in *Proc. IEEE Int'l Symp. Biomedical Imaging*, 2009, pp. 1039–1042.

[37] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua, "A fully automated

approach to segmentation of irregularly shaped cellular structures in EM images," in *Medical Image Computing and Computer-Assisted Intervention*, ser. Lecture Notes in Computer Science, vol. 6362, 2010, pp. 463–471.

[38] W.-K. Jeong, J. Beyer, M. Hadwiger, A. Vázquez-Reina, H. Pfister, and R. T. Whitaker, "Scalable and interactive segmentation and visualization of neural processes in EM datasets," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1505–1514, 2009.

[39] W.-K. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, A. Vázquez-Reina, C. Reid, J. W. Lichtman, and H. Pfister, "SSECRETT and NeuroTrace: Interactive visualization and analysis tools for large-scale neuroscience datasets," *IEEE Computer Graphics and Applications*, vol. 30, no. 3, pp. 58–70, 2010.

[40] E. Jurrus, R. Whitaker, B. Jones, R. Marc, and T. Tasdizen, "An optimal-path approach for neural circuit reconstruction," in *Proc. IEEE Int'l Symp. Biomedical Imaging*, 2008, pp. 1609–1612.

[41] E. Jurrus, M. Hardy, T. Tasdizen, P. Fletcher, P. Koshevoy, C.-B. Chien, W. Denk, and R. Whitaker, "Axon tracking in serial block-face scanning electron microscopy," *Medical Image Analysis*, vol. 13, no. 1, pp. 180–188, 2009.

[42] V. Kaynig, T. J. Fuchs, and J. M. Buhmann, "Geometrical consistent 3D tracing of neuronal processes in ssTEM data," in *Medical Image Computing and Computer-Assisted Intervention*, ser. Lecture Notes in Computer Science, 2010, vol. 6362, pp. 209–216.

[43] J. Lu, "Neuronal tracing for connectomic studies," *Neuroinformatics*, vol. 9, no. 2-3, pp. 1–8, 2011.

[44] E. Meijering, "Neuron tracing in perspective," *Cytometry Part A*, vol. 77A, no. 7, pp. 693–704, 2010.

[45] K. Saetzler, P. McCanny, E. P. Rodriguez, H. Horstmann, R. M. Bruno, and W. Denk, "A fuzzy algorithm to trace stained neurons in serial block-face scanning electron microscopy image series," in *Proc. Int'l Machine Vision and Image Processing Conf.*, 2009, pp. 162–167.

[46] S. N. Vitaladevuni and R. Basri, "Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2203–2210.

[47] J. H. Macke, N. Maack, R. Gupta, W. Denk, B. Schölkopf, and A. Borst, "Contour-propagation algorithms for semi-automated reconstruction of neural processes," *J. Neuroscience Methods*, vol. 167, no. 2, pp. 349–357, 2008.

[48] A. Vázquez-Reina, E. Miller, and H. Pfister, "Multiphase geometric couplings for the segmentation of neural processes," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 2020–2027.

[49] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, 1991.

[50] H.-F. Yang and Y. Choe, "3D volume extraction of densely packed cells in EM data stack by forward and backward graph cuts," in *Proc. IEEE Symp. Computational Intelligence for Multimedia Signal and Vision Processing*, 2009, pp. 47–52.

[51] Y. Kang, K. Engelke, and W. A. Kalender, "Interactive 3D editing tools for image segmentation," *Medical Image Analysis*, vol. 8, no. 1, pp. 35–46, 2004.

[52] V. Jain, H. S. Seung, and S. C. Turaga, "Machines that learn to segment images: a crucial technology for connectomics," *Current Opinion in Neurobiology*, vol. 20, no. 5, pp. 653–666, 2010.

[53] A. Elnakib, G. Gimel'farb, J. S. Suri, and A. El-Baz, "Medical image segmentation: A brief survey," in *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*. New York: Springer, 2011, pp. 1–39.

[54] D. L. Pham, C. Xu, and J. L. Prince, "A survey of current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, no. 315-337, pp. 315–338, 1998.

[55] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l J. Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[56] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[57] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations," *J. Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[58] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int'l J. Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[59] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-

initialization: A new variational formulation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. 430–436.

[60] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[61] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Trans. Image Processing*, vol. 17, no. 10, pp. 1940–1949, 2008.

[62] D. Xiao, W. S. Ng, C. B. Tsang, and U. R. Abeyratne, "A region and gradient based active contour model and its application in boundary tracking on anal canal ultrasound images," *Pattern Recognition*, vol. 40, no. 12, pp. 3522–3539, 2007.

[63] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[64] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, 2006.

[65] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *Proc. IEEE Int'l Conf. Computer Vision*, 2001, pp. 105–112.

[66] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *Int'l J. Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[67] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, 1984.

[68] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Secaucus, NJ: Springer-Verlag, 2001.

[69] J. M. Hammersley and P. Clifford, "Markov field on finite graphs and lattices," 1971.

[70] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.

[71] J. Besag, "On the statistical analysis of dirty pictures," *J. the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[72] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[73] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, 2006.

[74] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[75] L. R. Ford and D. R. Fulkerson, "A simple algorithm for finding maximal network flows and an application to the hitchcock problem," *Canadian J. Mathematics*, vol. 9, pp. 210–218, 1957.

[76] ——, *Flows in Networks*. Princeton, NJ: Princeton University Press, 1962.

[77] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[78] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[79] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008.

[80] P. Wattuya, K. Rothaus, J.-S. Praßni, and X. Jiang, "A random walker based approach to combining multiple segmentations," in *Proc. Int'l Conf. Pattern Recognition*, 2008, pp. 1–4.

[81] P. Kohli and P. H. S. Torr, "Measuring uncertainty in graph cut solutions - efficiently computing min-marginal energies using dynamic graph cuts," in *Proc. European Conf. Computer Vision*, 2006, pp. 30–43.

[82] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation," *IEEE Trans. Med. Imaging*, vol. 23, no. 7, pp. 903–921, 2004.

[83] N. I. Weisenfeld and S. K. Warfield, "Automatic segmentation of newborn brain MRI," *NeuroImage*, vol. 47, no. 2, pp. 564–572, 2009.

[84] T. Rohlfing, D. B. Russakoff, and C. R. M. Jr., "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *IEEE Trans. Med. Imaging*, vol. 23, no. 8, pp. 983–994, 2004.

[85] O. Commowick and S. K. Warfield, "Incorporating priors on expert performance parameters for segmentation validation and label fusion: A Maximum a Posteriori STAPLE," in *Medical Image Computing and Computer-Assisted Intervention*, ser. Lecture Notes in Computer Science, 2010, pp. 25–32.

[86] X. Artaechevarria, A. Muñoz-Barrutia, and C. O. de Solorzano, "Combination strategies in multi-atlas image segmentation: Application to brain MR data," *IEEE Trans. Med. Imaging*, vol. 28, no. 8, pp. 1266–1277, 2009.

[87] P. Coupé, J. V. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. L. Collins, "Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation," *NeuroImage*, vol. 54, no. 2, pp. 940–954, 2011.

[88] M. W. Jones and M. Chen, "A new approach to the construction of surfaces from contour data," *Comput. Graph. Forum*, vol. 13, no. 3, pp. 75–84, 1994.

[89] O. Nilsson, D. E. Breen, and K. Museth, "Surface reconstruction via contour metamorphosis: An eulerian approach with lagrangian particle tracking," in *Proc. IEEE Visualization*, 2005, pp. 407–414.

[90] H. Fuchs, Z. M. Kedem, and S. P. Uselton, "Optimal surface reconstruction from planar contours," *Commun. ACM*, vol. 20, no. 10, pp. 693–702, 1977.

[91] G. Barequet, D. Shapiro, and A. Tal, "Multilevel sensitive reconstruction of polyhedral surfaces from parallel slices," *The Visual Computer*, vol. 16, no. 2, pp. 116–133, 2000.

[92] C. L. Bajaj, E. J. Coyle, and K.-N. Lin, "Arbitrary topology shape reconstruction from planar cross sections," *Graphical Models and Image Processing*, vol. 58, no. 6, pp. 524–543, 1996.

[93] C. L. Bajaj and A. Gillette, "Quality meshing of a forest of branching structures," in *Proc. Int'l Meshing Roundtable*, 2008, pp. 433–449.

[94] J. Edwards and C. L. Bajaj, "Topologically correct reconstruction of tortuous contour forests," in *Proc. ACM Symp. Solid and Physical Modeling*, 2010, pp. 51–60.

[95] A. Vasilevskiy and K. Siddiqi, "Flux maximizing geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1565–1578, 2002.

[96] V. Kolmogorov and Y. Boykov, "What metrics can be approximated by geo-cuts, or global optimization of length/area and flux," in *Proc. IEEE Int'l Conf. Computer Vision*, 2005, pp. 564–571.

[97] N. Vu and B. S. Manjunath, "Graph cut segmentation of neuronal structures from transmission electron micrographs," in *Proc. Int'l Conf. Image Processing*, 2008, pp. 725–728.

[98] P. Dimitrov, C. Phillips, and K. Siddiqi, "Robust and efficient skeletal graphs," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 1417–1423.

[99] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. 755–762.

[100] M. van Ginkel, J. van de Weijer, L. J. van Vliet, and P. W. Verbeek, "Curvature estimation from orientation fields," in *Proc. Scandinavian Conf. Image Analysis*, 1999, pp. 545–551.

[101] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[102] R. Valenti, N. Sebe, and T. Gevers, "Image saliency by isocentric curvedness and color," in *Proc. IEEE Int'l Conf. Computer Vision*, 2009, pp. 2185–2192.

[103] A. Cardona, S. Saalfeld, P. Tomancak, and V. Hartenstein, "TrakEM2: open source software for neuronal reconstruction from large serial section microscopy data," in *Proc. High Resolution Circuits Reconstruction*, 2009, pp. 20–22.

[104] D. R. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, 2004.

[105] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[106] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction.* Springer, 2009.

[107] P. B. Bijari, A. Akhondi-Asl, and H. Soltanian-Zadeh, "Three-dimensional coupled-object segmentation using symmetry and tissue type information," *Computerized Medical Imaging and Graphics*, vol. 34, no. 3, pp. 236–249, 2010.

[108] T. Riklin-Raviv, N. Kiryati, and N. A. Sochen, "Segmentation by level sets and symmetry," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 1015–1022.

[109] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, 1967, pp. 362–380.

[110] S. Bouix, K. Siddiqi, and A. Tannenbaum, "Flux driven automatic centerline extraction," *Medical Image Analysis*, vol. 9, no. 3, pp. 209–221, 2005.

[111] M. S. Hassouna and A. A. Farag, "Variational curve skeletons using gradient vector flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2257–2274, 2009.

[112] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the poisson equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1991–2005, 2006.

[113] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, 2007.

[114] C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," in *Proc. European Conf. Computer Vision*, 2008, pp. 481–494.

[115] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 1605–1614.

[116] J. Rivest and P. Cabanagh, "Localizing contours defined by more than one attribute," *Vision Research*, vol. 36, no. 1, pp. 53–66, 1996.

[117] T. K. Leung and J. Malik, "Contour continuity in region based image segmentation," in *Proc. European Conf. Computer Vision*, 1998, pp. 544–559.

[118] L. Grady and G. Funka-Lea, "An energy minimization approach to the data driven editing of presegmented images/volumes," in *Medical Image Computing and Computer-Assisted Intervention*, ser. Lecture Notes in Computer Science, vol. 4191, 2006, pp. 888–895.

[119] N. Vu, "Image segmentation with semantic priors: A graph cut approach," Ph.D. dissertation, University of California at Santa Barbara, 2008.

[120] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 73–80.

[121] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3D scene priors for road detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 57–64.

[122] Y. J. Lee and K. Grauman, "Object-graphs for context-aware category discovery," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 1–8.

[123] J. M. Gonfaus, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez, "Harmony potentials for joint classification and segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 3280–3287.

[124] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3D brain image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1744–1757, 2010.

VITA

Huei-Fang Yang was born in Changhua, Taiwan. She received the degree of Bachelor of Education and Master of Education from National Taiwan Normal University in 1998 and 2001, respectively. After graduation, she taught computer science at National Tou-Lou Senior High School, Taiwan. She attended Texas A&M University as a graduate student in 2004 and received the Ph.D. degree in Computer Science in 2011.

The typist for this dissertation was Huei-Fang Yang.